Accelerations for Graph Isomorphism

Caishi Fang

Centre for Quantum Software and Information,
University of Technology Sydney, Australia

Abstract

In this paper, we present two main results. First, by only one conjecture (Conjecture 2.9) for recognizing a vertex symmetric graph, which is the hardest task for our problem, we construct an algorithm for finding an isomorphism between two graphs in polynomial time $O(n^3)$. Second, without that conjecture, we prove the algorithm to be of quasi-polynomial time $O(n^{1.5 \log n})$. The conjectures in this paper are correct for all graphs of size no larger than 5 and all graphs we have encountered. At least the conjecture for determining if a graph is vertex symmetric is quite true intuitively. We are not able to prove them by hand, so we have planned to find possible counterexamples by a computer. We also introduce new concepts like collapse pattern and collapse tomography, which play important roles in our algorithms.

1 Introduction

Currently, the best general algorithm for graph isomorphism problem is due to Babai [1], who shows that the graph isomorphism is of quasi-polynomial time $\exp((\log n)^{O(1)})$. We give a constructive proof of this result in the current paper. And by only one conjecture, which is quite true intuitively, we give a polynomial time algorithm for the problem. This also means that we have reduced the graph isomorphism problem in polynomial time to the problem of determining whether a graph is vertex symmetric or not.

A detailed review is needed on the applications and related problems such as group isomorphism. The author has not seen Babai’s [1] work in very detail. There may be some common techniques between this paper and previous papers not pointed out, which is another reason for the need of a review paper. Currently, if you want to know more about the origin and research history of the graph isomorphism problem, please refer to [8, 3, 1, 6].

In this paper, $G = (V, E)$ means an undirected graph $G$ with a vertex set $V$ and an edge set $E$ and without self-loop or multiple edges connecting two vertexes. The case that graphs containing self-loop and multi-edges is discussed after we have presented the main results. Let $V = \{v_1, v_2, \cdots, v_n\}$. If there is a direct connecting between $v_i$ and $v_j$ for $v_i, v_j \in V$ and $v_j \neq v_i$, we denote it by $(v_i, v_j)$. Let $T$ be the set of all $(i,j)$ pairs with $(v_i, v_j)$. $E = \{(v_i, v_j) : (i,j) \in T\}$.
We denote the size of $V$ as $|V|$. Let $V_j = \{ v_k \in V : (v_j, v_k) \in E \}$. The degree of a vertex $v_j \in V$ is $|V_j|$.

Given two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, if there is a one-to-one correspondence $\pi$ between $V_1$ and $V_2$, s.t., for all $v_i, v_j \in V_1$, $(v_i, v_j) \in E_1$ iff $(\pi(v_i), \pi(v_j)) \in E_2$, then we say $G_1$ is isomorphic to $G_2$ and $\pi$ is a graph isomorphism between $G_1$ and $G_2$. Any isomorphism from a graph $G$ to itself is called the automorphism of $G$. Note that, as the identity permutation is always an automorphism for any graph, we are not interested in this trivial automorphism. For the graph isomorphism and automorphism, we have an intuitive understanding, i.e., all directly connected vertexes must be also directly connected after an isomorphic or automorphic mapping, which is a permutation of vertex names.

Now we define several problems.

The graph isomorphism problem, denoted as $GI(G_1, G_2)$: determine whether $G_1$ and $G_2$ are isomorphic.

The graph automorphism problem, denoted as $GA(G)$: determine if there exists a non-trivial automorphism of the undirected graph $G$.

The graph automorphism counting problem, denoted as $\#GA(G)$: find the total number of automorphisms of the undirected graph $G$.

If we can solve a problem $P$ by using polynomially many times of the procedure for solving another problem $Q$, we say $P$ is polynomially reducible to $Q$. If $P$ and $Q$ are polynomially reducible to each other, we say they are polynomially equivalent.

It is shown $GI(G_1, G_2)$ is polynomially equivalent with $\#GA(G)$ [7, 5], and $GA(G)$ is polynomially reducible to $GI(G_1, G_2)$ [5].

It seems $\#GA(G)$ is much harder than $GA(G)$. Just take the complete graph as an example, the total number of automorphisms is $n! - 1$, which is hard to find one by one. With an oracle for $GI(G_1, G_2)$, it will be easy to do $\#GA(G)$.

In this paper, we introduce two more problems. One is the graph automorphism with constraint problem, denoted as $GA(G, C)$: determine if there exists a non-trivial automorphism of the undirected graph $G$, with the constraint $C$. For $v_1, v_2 \in G$, $(v_1, v_2) \in C$ if $v_1$ and $v_2$ cannot replace each other under any permutation or $\langle v_1, v_2 \rangle \in C$ if $v_1$ and $v_2$ can only correspond to each other in any automorphic mapping.

The other is the graph isomorphism with constraint problem, denoted as $GI(G_1, G_2, C)$: determine whether $G_1$ and $G_2$ are isomorphic, with the constraint $C$. For $v_1 \in G_1, v_2 \in G_2$, $(v_1, v_2) \in C$ if $v_1$ and $v_2$ cannot replace each other under any mapping or $\langle v_1, v_2 \rangle \in C$ if $v_1$ and $v_2$ can only correspond to each other in any isomorphic mapping.

It is easy to see that (1) $GI(G_1, G_2)$ is a special case of $GI(G_1, G_2, C)$; (2) $GA(G)$ is a special case of $GA(G, C)$; (3) $GI(G_1, G_2)$ is equivalent to $GA(G_1 \cup G_2, C)$, with $C = \{ \langle v_1, v_2 \rangle : v_1, v_2 \in G_1 \text{ or } v_1, v_2 \in G_2 \}$; (4) $GA(G, C)$ seems easier than $GA(G)$, as some permutations are ruled out by the constraint $C$. In our polynomial algorithm for $GI(G_1, G_2)$, we make use of $GI(G_1, G_2, C)$ as a subroutine.

In [7], it is said that the checking and counting of graph isomorphism are polynomially equivalent, which is an evidence to the conjecture that the graph isomorphism is not $NP$-complete.

In [2, 5], they define a complexity class $GI$ of all problems polynomially reducible to the graph isomorphism problem, and claim that $GI = P$ if the
graph isomorphism is in class $\mathbf{P}$.

We introduce concepts and theoretic work, including our conjectures, in Section 2. The algorithm for graph isomorphism is described in Section 3. The final section is the conclusion. The reader may skip Section 2.1 if not interested in too much theoretical work.

## 2 Preparations

Given two graphs $G = (V, E)$ and $G' = (V', E')$, with $|V| = |V'|$, our target is to find a one-to-one correspondence $\pi : V \to V'$, s.t.,

$$
\text{for all } v_i, v_j \in V, (v_i, v_j) \in E \text{ iff } (\pi(v_i), \pi(v_j)) \in E'.
$$

In this section, $V = \{v_1, v_2, \ldots, v_n\}$, $V' = \{v'_1, v'_2, \ldots, v'_n\}$; $V_i = \{v_j \in V : (v_i, v_j) \in E\}$, $V'_i = \{v'_j \in V' : (v'_i, v'_j) \in E'\}$, for $i = 1, 2, \ldots, n$.

The first information we can use is that $\pi(v_i) = v'_i$, i.e., in any isomorphism between $V$ and $V'$, a vertex of $V$ can only be mapped to a vertex of $V'$ with the same degree. For this reason, we introduce a concept called base subgraph $G^{(w)}$ for those vertexes of the same degree $w$ in a graph $G$:

$$
G^{(w)} = (V^{(w)}, E^{(w)}), \quad V^{(w)} = \{v_i \in V : |V_i| = w\},
$$

$$
E^{(w)} = \{(v_i, v_j) \in E : v_i, v_j \in V^{(w)}\}.
$$

Note that it is only possible to map a $v_i \in V^{(w)}$ to some $v'_i \in V'^{(w)}$.

Given $G^{(w)}$ and $v_i \in V^{(w)}$, we define the extension based on $v_i$,

$$
G^{ex}(v_i, w) = (V^{ex}(v_i, w), E^{ex}(v_i, w)),
$$

as follows:

1. $v_i \in V^{ex}(v_i, w)$;
2. $V_i - V^{(w)} \subseteq V^{ex}(v_i, w)$;
3. For any $v_j \in V^{ex}(v_i, w) - V^{(w)}$, $V_j \subseteq V^{ex}(v_i, w)$;
4. $E^{ex}(v_i, w) = \{(v_j, v_k) \in E : v_j \in V^{ex}(v_i, w) - V^{(w)}, v_k \in V^{ex}(v_i, w)\}$.

We call $v_i$ the base point of the extension $G^{ex}(v_i, w)$.

The base subgraph $G^{(w)}$ is a separation of the the graph $G$, which means vertexes inside the base subgraph are different from those outside. As such separation is not limited to the degree argument, we can generalize the concepts, base subgraph and extension, to a given set of vertexes $\beta \subseteq V$. The base subgraph of $\beta$ is $G^{\beta} = (V^{\beta}, E^{\beta})$, where $V^{\beta} = \beta$. The extension of $G^{\beta}$ based on $v_i$ is $G^{ex}(v_i, \beta)$. We just replace $V^{(w)}$ by $\beta$ in the definition of $G^{(w)}$ and $G^{ex}(v_i, w)$.

As the definitions are not so intuitive, we explain them by an example. Let’s consider the graph in Figure 1a. The base subgraph of degree 3 is depicted in Figure 1b. All extensions with respect to this base subgraph are depicted in Figure 1d, 1e. Note we have colored all base points in black.
We will learn more about the labeling procedure in the algorithm for graph isomorphism.

Now we introduce some sets of labels \( L_k = \{l_{k1}, l_{k2}, l_{k3}, \ldots \} \), for \( k = 1, 2, \ldots \). The total number of labels used in an algorithm will be finite. We use labels to replace extensions with respect to a base subgraph. The labels serve as carrying on the information of whether two extensions are isomorphic or not. If two extensions are isomorphic, we replace them with the same label, otherwise different labels. After pair-wise comparing Figure 1d, 1e, we find that they are not isomorphic with the constraint that a base point (black point) can only be mapped to a base point, so we label them differently. In Figure 1c, we focus on the base subgraph of degree 3, the base point \( v_2 \) now has a label \( l_{11} \) and the base point \( v_4 \) has a label \( l_{12} \). It is better to write labels of a base point in a predefined order and combine identical labels, e.g., write \( l_{23l_{11l_{12l_{12}}}l_{12}} \) as \( l_{11l_{12l_{12}}}l_{12} \).

We will learn more about the labeling procedure in the algorithm for graph isomorphism.

We say a graph is vertex regular of \( w \) if every vertex in the graph is of the same degree \( w \).

Given a graph \( G = (V, E) \), with \( v_i \) and \( V_i \) for \( i = 1, 2, \ldots, n \), we define the collapse of \( G \) with the trigger \( v_k \), \( G^{\text{col}}(v_k) \), as follows:

1. Layer 0: \( G^{\text{col}}(v_k, 0), V^{\text{col}}(v_k, 0) = \{v_k\}, E^{\text{col}}(v_k, 0) = \emptyset \);
2. Layer 1: \( G^{\text{col}}(v_k, 1), V^{\text{col}}(v_k, 1) = V_k, E^{\text{col}}(v_k, 1) = \{(v_i, v_j) \in E : v_i, v_j \in V_k\} \);
3. Layer \( i + 2 \): Given Layer \( i + 1 \) \( (G^{\text{col}}(v_k, i + 1)) \) and Layer \( i \) \( (G^{\text{col}}(v_k, i)) \), Layer \( i + 2 \) is \( G^{\text{col}}(v_k, i + 2), \) with \( V^{\text{col}}(v_k, i + 2) = \bigcup_{v_y \in V^{\text{col}}(v_k, i + 1)} \{v_y \in V : v_y \in V_x - V^{\text{col}}(v_k, i + 1) - V^{\text{col}}(v_k, i)\}, E^{\text{col}}(v_k, i + 2) = \{(v_x, v_y) \in E : v_x, v_y \in V^{\text{col}}(v_k, i + 2)\} \).

See Figure 2b for example. It is a collapse trigged by vertex \( v_1 \) for the vertex regular graph \( G_b \), which is Figure 2a. There are \( n \) collapses in a graph of \( n \) vertexes.

Let \( \text{deg}(v_i, v_j) = |V_i| + |V_j| - |V_i \cap V_j| \) be the degree (edge degree) of an edge \( (v_i, v_j) \). Intuitively, the degree of an edge is the total number of collapses in which the edge is located before Layer 2. Likewise, we say a graph is edge regular of \( w \) if every edge in the graph is of the same degree \( w \). Figure 3a...
and Figure 3b are both edge regular and vertex regular, and they are actually isomorphic. Figure 2a is vertex regular but not edge regular.

Figure 3: (a) graph $G_c$; (b) graph $G_d$

Let’s define here an important class of graphs, the vertex symmetric graph. For a graph $G = (V,E)$, by designating a nailed vertex (imagine that you nail the graph to the wall with the nail on the designated vertex), we distinguish $|V|$ possible nailed graphs of $G$, $G(v_i), i = 1, 2, \ldots, |V|$. Those independent parts of the graph will fall down to the floor. Thus, $G(v_i)$ is equal to $G^{col}(v_i)$ with the edges between its layers added back. Any nailed graph is connected. $G$ is vertex symmetric if and only if all of its nailed graphs are isomorphic to each other. More precisely, if for every $i$ from 1 to $|V|$, we have for every $j$ from $i$ to $|V|$, $GI(G(v_i), G(v_j), \{\langle v_i, v_j \rangle \}) = True$, then we say the graph $G$ is vertex symmetric. Here, $\{\langle v_i, v_j \rangle \}$ means $v_i$ can only correspond to $v_j$ in any isomorphic mapping and vice versa. Complete graph is vertex symmetric. A graph consisting of one circle or multiple identical circles is also vertex symmetric. Figure 3a and Figure 3b are further examples of vertex symmetric graphs. Later we will see that the only obstacle of the graph isomorphism problem is how to recognize a vertex symmetric graph.

**Lemma 2.1** Given $G = (V,E)$, if $G$ is vertex symmetric, then it is vertex regular.
We can generalize the concepts, collapse and nailed graph, to multi-collapse and multi-nailed graph, i.e., using multiple triggers or nailed vertexes at the same time. Given a graph $G$, a multi-collapse (sometimes just called collapse if it is clear from the content) with a set of triggers $T$ is $G^{\text{col}}(T)$, in which layer 0 contains vertexes in the set $T$, and layer $k + 1$ contains all vertexes connecting those vertexes in layer $k$. A multi-nailed graph with a set of nailed vertexes $T$ is $G(T)$, which looks the same as $G^{\text{col}}(T)$ with those edges between layers added back. The normal (not nailed) graph and the nailed graph are special cases of the multi-nailed graph.

A graph $G$ is edge symmetric if and only if all of its multi-nailed graphs $G((v_i, v_j))$, with $(v_i, v_j) \in E$, are isomorphic to each other under the constraint $\{\langle v_i, v_i' \rangle, \langle v_j, v_j' \rangle \}$, which means that the edge $(v_i, v_j)$ can only be mapped to $(v_i', v_j')$ in any isomorphism between $G((v_i, v_j))$ and $G((v_i', v_j'))$.

**Lemma 2.2** Given $G = (V, E)$, if $G$ is edge symmetric, then it is edge regular.

Actually, there exist graphs that is both vertex regular and edge regular but not vertex symmetric, see Figure 4a. There exist graphs that is edge symmetric but not vertex symmetric, e.g., Figure 4a, the complete bipartite graph in Figure 4b and Figure 3.2 in [4]. There exist graphs that is vertex symmetric but not edge symmetric, see Figure 4c. There exist graphs edge symmetric but not vertex regular, see Figure 4b. There exists graphs that is both vertex regular and edge regular but not vertex symmetric or edge symmetric, e.g., a graph consisting of two independent circles of different sizes.

![Figure 4: (a) graph $G_e$; (b) graph $G_f$; (c) graph $G_g$](image)

An arc is an edge treated as directed. A graph $G$ is arc symmetric if and only if all of its arc nailed graphs $G((v_i, v_j))$, with $(v_i, v_j) \in E$, are isomorphic to each other under the constraint $\{(v_i, v_i'), (v_j, v_j')\}$, which means that the vertex $v_i$ of the edge $(v_i, v_j)$ can only be mapped to $v_i'$ of the edge $(v_i', v_j')$ and $v_j$ can only be mapped to $v_j'$ in any isomorphism between $G((v_i, v_j))$ and $G((v_i', v_j'))$. The difference between arc symmetry and edge symmetry is that we treat the trigger edge as directed in the arc symmetry. There exists graphs that is both vertex symmetric and edge symmetric but not arc symmetric, e.g., the Doyle-Holt graph, see Figure 3.3 in [4].
Lemma 2.3 Given $G = (V, E)$, if $G$ is arc symmetric, then it is both vertex symmetric and edge symmetric.

In the literature, vertex symmetric is equivalent to vertex transitive and 0-arc transitive, edge symmetric is equivalent to edge transitive, and arc symmetric is equivalent to arc transitive and 1-arc transitive. We prefer the word ‘symmetric’ more than ‘transitive’, as the former is shorter and more intuitive.

Let’s point out the levels of symmetry of a graph: No pair of isomorphic vertex nailed subgraphs and no pair of isomorphic edge nailed subgraphs \(\Rightarrow\) Some pairs of isomorphic vertex nailed subgraphs or some pairs of isomorphic edge nailed subgraphs \(\Rightarrow\) Vertex symmetric or edge symmetric \(\Rightarrow\) Arc symmetric.

Given two multi-sets $A = \{a_1, a_2, \ldots, a_n\}$ and $B = \{b_1, b_2, \ldots, b_n\}$, we say $A$ matches $B$ if the sort of $a_1, a_2, \ldots, a_n$ is equal to the sort of $b_1, b_2, \ldots, b_n$. For instance, \(\{3, 7, 3, 2, 1\}\) matches \(\{1, 2, 3, 3, 7\}\) but does not match \(\{3, 7, 2, 1, 4\}\).

More generally, \(\{\{5, 3, 3\}, \{5, 2, 2, 8\}, \{1, 4, 2\}, \{1, 3, 3\}\}\) matches \(\{\{1, 2, 4\}, \{1, 3, 3\}, \{2, 2, 5, 8\}, \{3, 3, 5\}\}\) but does not match \(\{\{5, 3, 3\}, \{1, 3, 4\}, \{5, 2, 2, 2\}, \{1, 4, 2\}\}\).

The vertex property of a graph $G = (V, E)$ is the multi-set $P_v(G) = \{|V_i|: i = 1, 2, \ldots, n\}$, where $n = |V|$. We say two graphs $G$ and $G'$ are of the same vertex property if their vertex properties match. Similarly, the edge property of a graph $G = (V, E)$ is the multi-set $P_e(G) = \{\deg(v_i, v_j): (v_i, v_j) \in E\}$. We say two graphs $G$ and $G'$ are of the same edge property if their edge properties match.

Given one collapse of $G$, say $G_{\text{col}}(v_k)$, which has $l + 1$ layers, $G_{\text{col}}(v_k, i)$ for $i = 0, 1, \ldots, l$, the collapse tomography of $G_{\text{col}}(v_k)$ is an ordered list of $l$ ordered pairs of the vertex property and the edge property, i.e.,

$$C_{\text{tom}}(G, v_k) = \left[[P_v(G_{\text{col}}(v_k, i)), P_e(G_{\text{col}}(v_k, i))]: i = 1, 2, \ldots, l\right].$$

Two collapse tomographies match if all properties in the ordered list match at the corresponding position. For instance,

$$[[\{5, 3, 3\}, \{5, 2, 2, 8\}], [[1, 4, 2]; \{1, 3, 3\}], [[7, 5]; \{3, 3\}]]$$

matches

$$[[\{3, 3, 5\}, \{2, 2, 5, 8\}], [[1, 2, 4]; \{1, 3, 3\}], [[5, 7]; \{3, 3\}]]$$

but not

$$[[\{3, 3, 5\}, \{2, 2, 5, 8\}], [[1, 3, 3]; \{1, 2, 4\}], [[5, 7]; \{3, 3\}]].$$

The collapse tomography of Figure 2b is

$$[[\{0, 2, 2, 2\}; \{3, 3, 3\}], [[2, 2, 2]; \{3, 3, 3\}]].$$

Given a graph $G$, the collapse pattern of $G$ is a multi-set of collapse tomographies of $G$, i.e.,

$$C_{\text{pat}}(G) = \{C_{\text{tom}}(G, v_i) : i = 1, 2, \ldots, |V|\}.$$
Given a multi-nailed graph $G(T)$ and its collapse $G^{\text{col}}(T)$, which has $l + 1$ layers, the corresponding collapse pattern is

$$C^{\text{pat}}(G(T)) = \{C^{\text{tom}}(G, v_i) : v_i \in V^{\text{col}}(T, j) \} : j = 0, 1, 2, \ldots, l\}.$$  

If each vertex $v$ of the graph $G$ is labeled by $L(v)$, then the collapse tomography under labeling $L$ is

$$C^{\text{tom}}(G, L, v_i) = [L(v_i); P_V(G^{\text{col}}(v_k, i)); P_E(G^{\text{col}}(v_k, i))] : i = 1, 2, \ldots, l],$$

and the collapse pattern under labeling $L$ for a nailed graph $G(v_a)$ is

$$C^{\text{pat}}(G(v_a), L) = \{C^{\text{tom}}(G, L, v_i) : v_i \in V^{\text{col}}(v_a, j) \} : j = 0, 1, 2, \ldots, l\}.$$  

We leave the study of properties of the collapse pattern of normal graph, nailed graph and multi-nailed graphs as a future work.

**Lemma 2.4** Given $G = (V, E)$ and $G' = (V', E')$, if $G$ is isomorphic to $G'$, then the collapse pattern of $G$ matches that of $G'$.

**Conjecture 2.5** Given $G = (V, E)$ and $G' = (V', E')$, $G$ is isomorphic to $G'$ if the collapse pattern of $G$ matches that of $G'$.

**Lemma 2.6** Given $G$, $G'$, and their nailed graphs, $G(v_a)$ and $G'(v_b)$, if $G(v_a)$ is isomorphic to $G'(v_b)$ with the constraint \{\{v_a, v_b\}\}, then the collapse pattern of $G(v_a)$ matches that of $G'(v_b)$.

**Conjecture 2.7** Given $G$, $G'$, and their nailed graphs, $G(v_a)$ and $G'(v_b)$, $G(v_a)$ is isomorphic to $G'(v_b)$ with the constraint \{\{v_a, v_b\}\} if the collapse pattern of $G(v_a)$ matches that of $G'(v_b)$.

**Lemma 2.8** Given $G = (V, E)$, if $G$ is vertex symmetric, then all $|V|$ collapse tomographies of $G$ match each other, i.e., the collapse pattern of a vertex symmetric graph consists of $|V|$ equal collapse tomographies.

**Conjecture 2.9** Given $G = (V, E)$, $G$ is vertex symmetric if all $|V|$ collapse tomographies of $G$ match each other, i.e., the collapse pattern of a vertex symmetric graph consists of $|V|$ equal collapse tomographies.

Conjecture 2.9 is a corollary of Conjecture 2.7. In the algorithm for graph isomorphism, we only use Conjecture 2.9, as it is quite true intuitively.

Let’s call a graph $G$ *vertex indistinguishable* if all of its $|V|$ collapse patterns $C^{\text{pat}}(G(v_i))$, with $v_i \in V$, are equivalent. The question corresponding to Conjecture 2.9 is “Is there any graph that is vertex indistinguishable but not vertex symmetric?”.

**Lemma 2.10** Given $G = (V, E)$, if $G$ is edge symmetric, then all $|E|$ collapse patterns of edge nailed graphs $G(v_i, v_j)$, with $(v_i, v_j) \in E$, match each other.

**Conjecture 2.11** Given $G = (V, E)$, $G$ is edge symmetric if all $|E|$ collapse patterns of edge nailed graphs $G(v_i, v_j)$, with $(v_i, v_j) \in E$, match each other.
In order to have a similar conjecture for arc symmetry, we redefine the collapse pattern for the arc nailed graph $G(v_a; v_b)$ as an ordered list of collapse patterns:

$$C_{\text{pat}}(G(v_a; v_b)) = [C_{\text{pat}}(G(v_a, v_b)), C_{\text{pat}}(G[v_a, v_b](v_a)), C_{\text{pat}}(G[v_a, v_b](v_b))]$$

where $C_{\text{pat}}(G(v_a, v_b))$ is the collapse pattern of the edge nailed graph $G(v_a, v_b)$, $C_{\text{pat}}(G[v_a, v_b](v_a))$ is the collapse pattern of the nailed graph at $v_a$ obtained by removing the edge $(v_a, v_b)$ in $G$ and nail the vertex $v_a$, and similarly for $C_{\text{pat}}(G[v_a, v_b](v_b))$.

**Lemma 2.12** Given $G = (V, E)$, if $G$ is arc symmetric, then all $2|E|$ collapse patterns of arc (an edge treated as directed) nailed graphs $G(v_i; v_j)$, with $(v_i, v_j) \in E$ and the direction is from $v_i$ to $v_j$, match each other.

**Conjecture 2.13** Given $G = (V, E)$, $G$ is arc symmetric if all $2|E|$ collapse patterns of arc nailed graphs $G(v_i; v_j)$ match each other.

The quantity, collapse pattern, contains a lot of information of a graph, so it may be a good argument to distinguish graphs. Up to now, our conjectures have never failed. Although we have tried our best to prove them, we cannot. Thus, we hope of finding a counterexample by a computer.

Now let’s introduce the dual graph. Given a graph $G = (V, E)$, its dual graph is $\overline{G} = (\overline{V}, \overline{E})$, where $(v_i, v_j) \in \overline{E}$ if and only if $(v_i, v_j) \notin E$.

**Lemma 2.14** $G$ is isomorphic to $G'$ if and only if $\overline{G}$ is isomorphic to $\overline{G}'$.

**Proof.** For any isomorphism $\pi$ between $G$ and $G'$, $(v_i, v_j)$ in $G$ if and only if $(\pi(v_i), \pi(v_j))$ in $G'$. So it is also true that $(v_i, v_j)$ not in $G$ if and only if $(\pi(v_i), \pi(v_j))$ not in $G'$.

When a graph of size $n$ is vertex regular of a large degree $w > \frac{n}{2}$, it seems easier to consider its dual graph, which is vertex regular of a smaller degree $n - w < \frac{n}{2}$.

**Lemma 2.15** Given two complete graphs $G$ and $G'$, if $G$ has $n$ vertexes with vertex $v_i$ labeled by $L(v_i)$ and $G'$ has $n$ vertexes with vertex $v_i'$ labeled by $L(v_i')$, then $G$ is isomorphic to $G'$ if and only if $\{L(v_i) : i = 1, 2, \ldots, n\}$ matches $\{L(v_i') : j = 1, 2, \ldots, n\}$.

**Proof.** The dual graph of a complete graph of size $n$ is $n$ independent vertexes.□

### 2.1 Variations of Collapse Pattern

In the remaining part of Section 2, we provide more discussions on our conjectures, which is not important for the next section. The reader may skip it if not interested.

In case that the above conjectures fail, we can vary the definitions of our collapse pattern to add more details of the graph.

Given a nailed graph $G(v_a; v_b)$, let $G^{\text{col}}(v_a, j)$, $j = 0, 1, 2, \ldots, l$, be $l + 1$ layers of its collapse, and let $G^{\text{col}}(v_i; v_a, j)$ be the extension with a base point $v_i$ in
Lemma 2.17 If \( v \) are well-defined and computable in polynomial time.

All of our definitions of collapse tomography and collapse pattern are well-defined and computable in polynomial time.

Lemma 2.16 All of our definitions of collapse tomography and collapse pattern are well-defined and computable in polynomial time.

\[
C_{\text{pat}}(G) = \{ C_{\text{pat}}(G(v_i)) : v_i \in V \}.
\]

If \( G \) be a multi-nailed graph with nailed vertexes \( T \) and its collapse \( G^{\text{col}}(T) \), which has \( l + 1 \) layers. The corresponding varied collapse pattern of \( G(T) \) is

\[
C_{\text{pat}}^{\text{var}}(G(T)) = \{ [C_{\text{tom}}^G(G, v_i); C_{\text{tom}}^G(G^{\text{col}}(T, j), v_i); C_{\text{tom}}^G(G^{\text{ex}}(v_i; T, j), v_i)] : v_i \in V^{\text{col}}(T, j)] : j = 0, 1, \ldots, l \}.
\]

This varied definition encodes more information of the nailed graph.

Lemma 2.17 If \( G \) is a diverging graph with the source \( v \) and \( n - 1 \) vertexes labeled by \( L(v_i) \), and \( G' \) is another diverging graph with source \( v' \) and \( m - 1 \) vertexes labeled by \( L(v'_j) \), then \( G(v) \) is isomorphic to \( G'(v') \) if and only if \( \{ L(v_i) : v_i \in V \} \) matches \( \{ L(v'_j) : v'_j \in V' \} \).

Theorem 2.18 Given \( G(v_a) \) and \( G'(v_b) \), \( G(v_a) \) is isomorphic to \( G'(v_b) \) with the constraint \( \{ (v_a, v_a) \} \) if and only if \( C^q(G(v_a)) \) matches \( C^q(G'(v_b)) \), if we define the quantity as

\[
C^q(G(v_a)) = \{ C^q(G^{\text{ex}}(v_i, v_a)) : v_i \in V_a \},
\]

where \( G^{\text{ex}}(v_i, v_a) \) is the extension based on \( v_i \) linking to \( v_a \) and the base subgraph consisting only of the edge \( (v_i, v_a) \).

Proof. Let \( L(v_i) = C^q(C^{\text{ex}}(v_i, v_a)) \), then by induction and Lemma 2.17, we can prove this theorem. \( \square \)

Corollary 2.19 If \( C^q(G(v_a)) \) is a quantity for nailed graphs, then \( C^q(G) = \{ C^q(G(v_i)) : v_i \in V \} \) is a quantity for normal graphs.

Lemma 2.20 If \( C^q(G) = \{ C^q(G(v_i)) : v_i \in V \} \) is a quantity for normal graphs, then \( C^q(G, L) = \{ [C^q(G(v_i)), L(v_i)] : v_i \in V \} \) is a quantity for the case that each vertex is labeled by \( L(v_i) \).
Lemma 2.21 If $G(v_a)$ and $G'(v_b)$ are two nailed graphs with their collapses $G^{col}(v_a)$ and $G'^{col}(v_b)$ having only Layer 0 and Layer 1, then $G(v_a)$ is isomorphic to $G'(v_b)$ if and only if $G^{col}(v_a, 1)$ is isomorphic to $G'^{col}(v_b, 1)$.

Theorem 2.22 Given $G(v_a)$ a nailed graph, if we define the quantity as

$$C^q(G(v_a)) = \{[C^q(G(v_i, V_a)), C^q(G^{ex}(v_i, V_a \cup \{v_a\}))]: v_i \in V^{col}(v_a, 1)\},$$

where $G(v_i, V_a)$ is the graph $G^{col}(v_a, 1)$ nailed at $v_i$, and $G^{ex}(v_i, V_a \cup \{v_a\})$ is the extension based on $v_i$ with the base subgraph consisting of vertexes $V_a$ and $v_a$, then this quantity can distinguish graphs according to isomorphism.

Proof. Let $L(v_i) = C^q(G^{ex}(v_i, V_a \cup \{v_a\}))$, then by induction and Lemma 2.20 and Lemma 2.21, we can prove this theorem.

As the lack of details of a graph, we cannot accelerate the graph isomorphism problem in the abstract approach. In the next section, we apply a bunch of methods for the acceleration of our algorithm.

3 Algorithms

Now, we are ready to present the algorithm for determining whether two graphs are isomorphic.

Note that our algorithm is only based on Conjecture 2.9 and no other assumption not proved. If we remove this conjecture, we can show its quasi-polynomial time efficiency.

Our algorithm for graph isomorphism is $GI(G, G') = GI(G, G', L_0)$, where $L_0 = \emptyset$.

Algorithm: Graph Isomorphism $GI(G, G', L_k)$ with Labels

Input: Two graphs $G = (V, E)$ and $G' = (V', E')$, without self-loop or multi-edge. $V = \{v_1, v_2, \ldots, v_n\}$, $V' = \{v'_1, v'_2, \ldots, v'_n\}$. Every vertex $v \in V \cup V'$ is labeled by $L_k(v)$.

Output: If $G$ and $G'$ are isomorphic under the labeling function $L_k$, output ‘Yes’, otherwise ‘No’.

Runtime: $O(n^3)$ with Conjecture 2.9; $O(n^{1.5 \log n})$ without Conjecture 2.9.

The worst case is when the input is two vertex indistinguishable but not vertex symmetric graphs.

Procedure:

1. Compute $C^{tom}(G, L_k, v_i)$ for all $v_i \in V$, and $C^{tom}(G', L_k, v'_i)$ for all $v'_i \in V'$. If they all match each other, then by Conjecture 2.9, $G$ and $G'$ are vertex symmetric. Then if $G$ and $G'$ are regular of degree $w \leq \frac{n}{2}$, we call the sub-algorithm $GI(G, G', C, L_k)$, with $C = \{v_1, v'_1\}$. If $G$ and $G'$ are regular of degree $w > \frac{n}{2}$, we call the sub-algorithm $GI(G', G, C, L_k)$, with $C = \{v_1, v'_1\}$. Output ‘Yes’ if the sub-algorithm returns ‘Yes’. Here, our target is to find an isomorphism. Note that, if we do not appeal
to Conjecture 2.9, then we have to check the isomorphism for all $2n$ nailed graphs of $G$ and $G'$ in the worst case, instead of only one pair. We may use the method ‘try-and-error’ to further accelerate the algorithm, as we believe that there is a non-negligible probability for finding a pair of isomorphic nailed graphs if two graphs are vertex indistinguishable but not vertex symmetric. Because of the use of dual graph, the size of the next occurrence of vertex indistinguishable graph is no larger than $\frac{n}{2}$.

2. Compute $C(G) = \{C^{pat}(G(v_i), L_k) : v_i \in V\}$, and $C(G') = \{C^{pat}(G'(v'_j), L_k) : v'_j \in V'\}$. Sort these two multi-sets of collapse patterns. If they do not match, output ‘No’. Otherwise, suppose the rarest collapse pattern in $C(G)$ is $C^{pat}(G(v_a), L_k)$ and the corresponding one in $C(G')$ is $C^{pat}(G'(v'_b), L_k)$, let $\beta = \{v_i : C^{pat}(G(v_i), L_k) = C^{pat}(G(v_a), L_k)\}$ and $\beta' = \{v'_j : C^{pat}(G'(v'_j), L_k) = C^{pat}(G'(v'_b), L_k)\}$. The base subgraph of $G$ is $G^\beta$ and that of $G'$ is $G'^{\beta'}$. The extensions of $G^\beta$ are $Ex(\beta) = \{G^\beta(v_i, \beta) : v_i \in V^\beta\}$ and those of $G'^{\beta'}$ are $Ex(\beta') = \{G'^{\beta'}(v'_j, \beta') : v'_j \in V'^{\beta'}\}$. Call the sub-algorithm $GI(G^\beta(v_1, \beta_1), G'^{\beta'}(v_2, \beta_2), C, L_k)$ for all pairs of extensions $(G^\beta(v_1, \beta_1), G'^{\beta'}(v_2, \beta_2))$ in the union set $Ex(\beta) \cup Ex(\beta')$ with $C = \{(v_1, v_2)\}$. Next we assign two extensions with the same label iff they are isomorphic. The labeling function $L_{k+1}$ assigns two base points with the same label iff $L_k$ assigns them with the same label and the labels for their extensions are of the same. Then we call the sub-algorithm $GI(G^\beta, G'^{\beta'}, L_{k+1})$, with vertex $v \in \beta \cup \beta'$ labeled by $L_{k+1}(v)$. Output ‘Yes’ iff this sub-algorithm returns ‘Yes’.

**Algorithm: Graph Isomorphism GI(G, G', C, L_k) with Constraint and Labels**

**Input:** Two graphs $G = (V, E)$ and $G' = (V', E')$, without self-loop or multi-edge. $V = \{v_1, v_2, \cdots, v_n\}$, $V' = \{v'_1, v'_2, \cdots, v'_n\}$. The constraint is $C = \{(v_a, v'_b)\}$, where $v_a$ is the nailed vertex of $G(v_a)$ and $v'_b$ is the nailed vertex of $G'(v'_b)$. Every vertex $v \in V \cup V'$ is labeled by $L_k(v)$.

**Output:** If $G(v_a)$ and $G'(v'_b)$ are isomorphic under the constraint $C$ and the labeling function $L_k$, then output ‘Yes’; otherwise ‘No’.

**Procedure:**

1. Compute $C^{pat}(G(v_a), L_k)$ and $C^{pat}(G'(v'_b), L_k)$. If they do not match, then output ‘No’. Otherwise, let the collapse $G^{col}(v_a)$ have $l + 1$ layers.

2. If $l = 1$, call the sub-algorithm $GI(G^{col}(v_a), 1, G'^{col}(v'_b), 1, L_k)$. Output ‘Yes’ iff this sub-algorithm returns ‘Yes’. Here, the input graphs may be again vertex indistinguishable but not vertex symmetric.

3. For $l \geq 1$, from $x = l - 1$ to $x = 0$, we update the labels of vertexes in layer $x$ with the layer $x + 1$ by the following procedure. Let $\beta(x)$ be the set of vertexes in layers from 0 to $x$ of $G^{col}(v_a)$ and $\beta'(x)$ be the set of vertexes in layers from 0 to $x$ of $G'^{col}(v'_b)$. Let $Ex(\beta(x)) = \{G^{\beta}(v_i, \beta(x)) : v_i \in V^{\beta}(v_a, x)\}$ and $Ex(\beta'(x)) = \{G'^{\beta'}(v'_j, \beta'(x)) : v'_j \in V'^{\beta'}(v'_b, x)\}$.
Suppose the labeling function for layer $x$ is $L_{k,x}$ with $L_{k,1} = L_k$. Call the sub-algorithm $GI(G_1^x(v_1, \beta_1(x)), G_2^x(v_2, \beta_2(x)), C, L_{k,x+1})$ for all pairs of extensions $(G_1^x(v_1, \beta_1(x)), G_2^x(v_2, \beta_2(x)))$ in the union set $Ex(\beta(x)) \cup Ex(\beta'(x))$ with $C = \{(v_1, v_2)\}$. Usually, we omit the labels of base points when we are determining isomorphism of their extensions. Next we assign two extensions with the same label iff they are isomorphic. The labeling function $L_{k,x}$ assigns two base points with the same label iff $L_k$ assigns them with the same label and the labels for their extensions are of the same. On the whole, we distinguish two nailed graphs, according to isomorphism, layer-by-layer.

**Theorem 3.1** The algorithm $GI(G, G', L_k)$ can correctly distinguish graphs according to isomorphism and is of polynomial time if Conjecture 2.9 is true.

**Theorem 3.2** If Conjecture 2.9 is not true, then the algorithm $GI(G, G', L_k)$ is of quasi-polynomial time $O(n^{1.5 \log n})$.

**Proof.** The recursion is $T(n) = mn^2T(\frac{n}{2})$ for the worst case, which has the most chance of encounter of graphs that is vertex indistinguishable but not vertex symmetric. \[\square\]

Suppose now we allow graphs with self-loop and multi-edge. First, we remove all self-loop and multi-edge, and by the algorithm $GI(G, G')$, we can find an isomorphism. Then we can check this isomorphism for graphs with self-loop and multi-edge. Another method is to use the labeling procedure to remove self-loop and multi-edge before the algorithm for graphs without those.

With the algorithm for isomorphism, we can construct an algorithm for automorphism without too much effort.

4 Conclusion

In this paper, we provide an algorithm for the graph isomorphism problem. We have shown that the graph isomorphism problem is based on the recognition problem of a vertex symmetric graph. Conjecture 2.9 can solve the latter problem very efficiently. Without this conjecture, our algorithm is of quasi-polynomial time. It is possible to further improve our algorithm.

Although some techniques in this article might have been already known before, we apologize for the possibility of not pointing out, for our limited knowledge. We plan to write a detailed and also interesting review paper to make it clear and talk all aspects of the graph isomorphism problem, including the research history and its applications.

Left works: What is the algorithm for directed graphs? Experimental benchmark of algorithms for graph isomorphism. Problems relating to the graph isomorphism. Properties of collapse tomography and collapse pattern. A more refined analysis of our algorithm. Find a counter-example to Conjecture 2.9 and other conjectures.
References

[1] László Babai. Graph isomorphism in quasipolynomial time [extended abstract]. In Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, pages 684–697. ACM, 2016.

[2] Kellog S Booth and Charles J Colbourn. Problems polynomially equivalent to graph isomorphism. Computer Science Department, Univ., 1979.

[3] Horst Bunke. Graph matching: Theoretical foundations, algorithms, and applications. In Proc. Vision Interface, volume 2000, pages 82–88, 2000.

[4] Chris Godsil and Gordon Royle. Algebraic graph theory. Springer-Verlag, 2001.

[5] J Köbler, U Schöning, and J Torán. The Graph Isomorphism Problem: Its Structural Complexity. Birkhaeuser, Boston, MA, 1993.

[6] Johannes Köbler, Uwe Schöning, and Jacobo Torán. The graph isomorphism problem: its structural complexity. Springer Science & Business Media, 2012.

[7] Rudolf Mathon. A note on the graph isomorphism counting problem. Information Processing Letters, 8(3):131–136, 1979.

[8] Gary L Miller. Graph isomorphism, general remarks. Journal of Computer and System Sciences, 18(2):128–142, 1979.