Maximizing Connections: The co-NP-Completeness of Identifying the Largest Jigsaw Puzzle from a Set of Pieces

Sumedh Venuganti

December 26, 2022

Abstract

It is proven that given n jigsaw puzzle pieces, determining the maximum number of puzzle pieces that connect is NP-hard. This problem belongs to co-NP and is co-NP-complete.

1 Introduction

Four months ago, professional machinist Shane Wighton released a video about building a robot that can solve a large jigsaw puzzle on his YouTube channel Stuff Made Here. Professor Nickvash Kani from University of Illinois Urbana-Champaign offered his students a $3,000,000,000 cash reward, a retroactive A+ in his class, and most importantly an RA position, for answering if the jigsaw-puzzle discussed in the video belongs to P, NP, NP-hard, NP-complete, or co-NP. Suppose there are n puzzle pieces in that may or may not belong to the same puzzle. What is the maximum number of puzzle pieces k that can be connected? This paper proves that determining k is co-NP-complete.

2 Classifying the Complexity of Jigsaw Puzzle Problems

MaxJigsaw accepts as input n distinctly shaped four-sided jigsaw puzzle pieces and determines the maximum number of pieces k that can be connected such that k ≤ n and every two pieces that interlock with one another share a compatible edge connection.

HasHamiltonianPath accepts as input a directed graph G = (V, E) with |V| vertices and |E| directed edges, and determines whether there exists a sequence of vertices and edges in G that forms a path that visits every vertex in V exactly once.

2.1 MaxJigsaw \notin NP

A problem instance of MaxJigsaw consists of n four-sided puzzle pieces. A certificate c consists of an integer k, the thought-to-be maximum number of pieces that can connect, and an assignment of placements to the k pieces. While a certifier algorithm C(I,c) can verify in polynomial time that the pieces in c are all used and that adjacent pieces in c have compatible connections, assuming P\neq NP, C(I,c) cannot verify in polynomial time that k is the maximum number of pieces that can be connected, as doing so would require solving the NP-hard MaxJigsaw problem itself to find the true maximum.

2.2 MaxJigsaw is NP-hard

MaxJigsaw is at least as hard as the (NP-complete) directed Hamiltonian Path problem. HasHamiltonianPath \leq_p MaxJigsaw

Figure 1: A decider for the directed Hamiltonian Path problem.
The NP-hardness of MaxJigsaw is shown by reduction from HasHamiltonianPath in polynomial time. Figure 1 outlines the process for solving HasHamiltonianPath using the oracle MaxJigsaw. An arbitrary instance of the Hamiltonian Path Problem \( G(V, E) \) is transformed using a polynomial time reduction, Transform, into a specific instance of the MaxJigsaw problem \( J_e \). The oracle MaxJigsaw returns the size of the largest set of puzzle pieces that can be connected \( k \). If \( k \geq 2|V| - 1 \), HasHamiltonianPath returns True. Otherwise, it returns False. Transform accepts as input \( G(V, E) \), where \( V \) is a set of distinct vertices labeled \( V = \{v_1, v_2, ..., v_n\} \) and \( E = \{(v_i, v_j) \mid \text{exists a directed edge from } v_i \text{ to } v_j\} \). For each edge piece and for each vertex piece, Transform\( (G(V, E)) \) creates a maximum of \( n = 2|V| + 2 \) indents and extrusions. Transform has a polynomial time complexity \( O(|V||E|) + O(|V|^2) = O(|V|^2) \).

\[\text{Transform}(G(V, E))\]

\[
\begin{align*}
\text{pieces} & \leftarrow \text{empty set of puzzle pieces} \\
\text{for each vertex } v_i \text{ in } V & \text{ do} \\
\text{~~~} & \text{p} \leftarrow \text{an } (n+2) \times (n+3) \text{ puzzle piece with all columns except the last one filled} \\
\text{~~~} & \text{add } i \text{ inwards squares to the left side of } p[2:2+a, 1] \\
\text{~~~} & \text{add } i \text{ outwards teeth to the right side of } p[2:2+b, n+3] \\
\text{~~~} & \text{add an inwards notch to the top edge of } p[1, \lfloor \frac{n+3}{2} \rfloor] \\
\text{~~~} & \text{add an inwards notch to the bottom edge of } p[n+2, \lfloor \frac{n+3}{2} \rfloor] \\
\text{~~~} & \text{pieces.add(p)} \\
\text{for each directed edge } (v_a, v_b) \text{ in } E & \text{ do} \\
\text{~~~} & \text{p} \leftarrow \text{an } (n+2) \times (n+3) \text{ puzzle piece with all columns except the last one filled} \\
\text{~~~} & \text{add } a \text{ inwards teeth to the left side of } p[2:2+a, 1] \\
\text{~~~} & \text{add } b \text{ outwards squares to the right side of } p[2:2+b, n+3] \\
\text{~~~} & \text{add an inwards notch to the top edge of } p[1, \lfloor \frac{n+3}{2} \rfloor] \\
\text{~~~} & \text{add an inwards notch to the bottom edge of } p[n+2, \lfloor \frac{n+3}{2} \rfloor] \\
\text{~~~} & \text{pieces.add(p)} \\
\text{return pieces}
\end{align*}
\]

Input graph \( G(V, E) \) in Figure 2 is a YES instance of the Hamiltonian Path problem. The problem instance \( J_e \), consisting of 11 puzzle pieces, is generated by passing \( G(V, E) \) into the Transform routine.

Figure 2: \( G(V, E) \) depicts an instance of the Hamiltonian Path problem that is passed as input into Transform\( (G(V, E)) \) to produce six jigsaw puzzle pieces \( J_e \), an instance of MaxJigsaw.
The puzzle pieces marked in red in Figure 3 correspond to vertices and edges in \( G(V,E) \) that form the Hamiltonian Path. For the problem instance \( J_e \) generated Figure 2, the oracle \textbf{MaxJigsaw} determines that a maximum of 9 consecutive puzzle pieces can be connected. Since \( 9 \geq |2|−1 \), \textbf{HasHamiltonianPath} returns True. The oracle is capable of determining the maximum number of 4-sided puzzle pieces that can connect in any placement as long as connections between all adjacent pieces are valid. The specific placement of squares, teeth, and notches imposed by \text{TRANSFORM} convert arbitrary directed graphs to specific puzzle pieces whose maximum connectivity is achieved by a kx1 or 1xk placement. A proof of correctness, justifying the validity of the reduction, is provided below.

**Necessity:** \( G(V,E) \) has a Hamiltonian path \( \Rightarrow \geq 2|V|−1 \) pieces in \( J_e \) are connected

\[ \Rightarrow \exists \text{ a finite sequence of distinct edges in } G \text{ that travel from one vertex to a different vertex in } G \text{ and visit each vertex in } G \text{ exactly once.} \]

\[ \Rightarrow \exists \text{ a sequence of distinct vertices and edges } v_{start}, e_{(start,i)}, v_i, e_{(i,j)}, \ldots, v_{fin}, \text{ comprising of } |V| \text{ vertices and } |V|−1 \text{ edges visiting visit every vertex in } G \text{ exactly once} \]

\[ \Rightarrow \forall v_i \in V, \exists \text{ a corresponding piece with } i \text{ left sided square indents } i \text{ right sided teeth extrusions} \]

\[ \Rightarrow \forall v_i, v_j \in V, \exists \text{ a piece with } i \text{ left sided teeth indents and } j \text{ right sided square extrusions} \]

\[ \Rightarrow \forall |V| \text{ vertices } v_{start}, e_{(start,i)}, v_i, e_{(i,j)}, \ldots, v_{fin}, \exists \text{ a corresponding piece in } J_e \text{ and } \forall |V|−1 \text{ edges} \]

\[ \Rightarrow \exists \text{ in } G \text{ connecting two vertices, } \exists \text{ a puzzle piece connecting the two corresponding vertex pieces} \]

\[ \Rightarrow \geq 2|V|−1 \text{ pieces in } J_e \text{ are connected} \]

**Sufficiency:** \( \geq 2|V|−1 \) pieces in \( J_e \) are connected \( \Rightarrow G(V,E) \) has a Hamiltonian path

- Vertex pieces in \( J_e \) have square indents on the left, teeth extrusions on the right, and notches on the top and bottom that prevent any two vertex pieces from being directly connected.
- Edge pieces in \( J_e \) have teeth indents on the left, square indents on the right, and notch indents on the top and bottom that prevent any two edge pieces from being connected.
- These geometric constraints make it impossible for two vertex pieces \( v_i \) and \( v_j \) to be connected unless there is an edge piece corresponding to an edge from \( v_i \) to \( v_j \) in \( G \).

\[ \geq 2|V|−1 \text{ pieces in } J_e \text{ are connected} \]

\[ \Rightarrow \text{ all vertices of } V \text{ are included somewhere in the sequence (due to geometric constraints)} \]

\[ \Rightarrow \text{ there are } |V|−1 \text{ edge pieces included in the sequence between the } |V| \text{ pieces} \]

\[ \Rightarrow \exists \text{ an alternating sequence of vertex and edge pieces of size the } 2|V|−1 \text{ including all vertex pieces} \]

\[ \Rightarrow \exists \text{ a corresponding alternating sequence of vertices and edges in } G \text{ that visits every vertex once} \]

\[ \Rightarrow G \text{ has a Hamiltonian path} \]

\[ \geq 2|V|−1 \text{ pieces in } J_e \text{ are connected} \]

\[ \Rightarrow \text{ all vertices of } V \text{ are included somewhere in the sequence (due to geometric constraints)} \]

\[ \Rightarrow \text{ there are } |V|−1 \text{ edge pieces included in the sequence between the } |V| \text{ pieces} \]

\[ \Rightarrow \exists \text{ an alternating sequence of vertex and edge pieces of size the } 2|V|−1 \text{ including all vertex pieces} \]

\[ \Rightarrow \exists \text{ a corresponding alternating sequence of vertices and edges in } G \text{ that visits every vertex once} \]

\[ \Rightarrow G \text{ has a Hamiltonian path} \]

2.3 \textbf{MaxJigsaw} \( \in \text{ co-NP} \)

A NO instance \( I \) of \textbf{MaxJigsaw} consists of (1) a subset of \( n \) four-sided puzzle pieces that have at least two puzzle pieces that have an incompatible connection, or (2) a subset of \( n \) four-sided puzzle pieces have valid connections for adjacent pieces for some integer less than the true maximum number of pieces that can be connected.

In Case 1, a certificate \( c \) consists of an integer \( k \), the thought-to-be maximum of the \( n \) puzzle pieces that can be connected, and an assignment of placements to the \( k \) pieces. A certifier algorithm \( C(I,c) \) can parse the certificate \( c \) and verify in polynomial time at least one set of adjacent pieces in \( c \) have incompatible edge connections or that \( k \) pieces are used.

In Case 2, a certificate \( c \) consists of (1) an integer \( q \) smaller than the true maximum number of puzzle pieces that can connect, (2) an integer \( k' > q \) such that \( k' \) puzzle pieces connect, and (3) an assignment of placements to the \( k' \) pieces. A certifier algorithm \( C(I,c) \) can parse the certificate \( c \) and verify in polynomial time that every set of two adjacent pieces from \( k' \) are connected with a compatible edge. \( C(I,c) \) can verify the correctness of the counterexample provided in \( c \) in polynomial time.
2.4 MaxJigsaw $\not\in$ NP-complete

Since MaxJigsaw $\not\in$ NP, it is not NP-complete.

2.5 MaxJigsaw $\in$ co-NP-complete

MaxJigsaw $\in$ co-NP and is NP-hard, thus it is co-NP-complete.

References

