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

Sumedh Vemuganti

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 $k \leq n$ 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!=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.

$$\textbf{HasHamiltonianPath} \leq_{p} \textbf{MaxJigsaw} \tag{1}$$

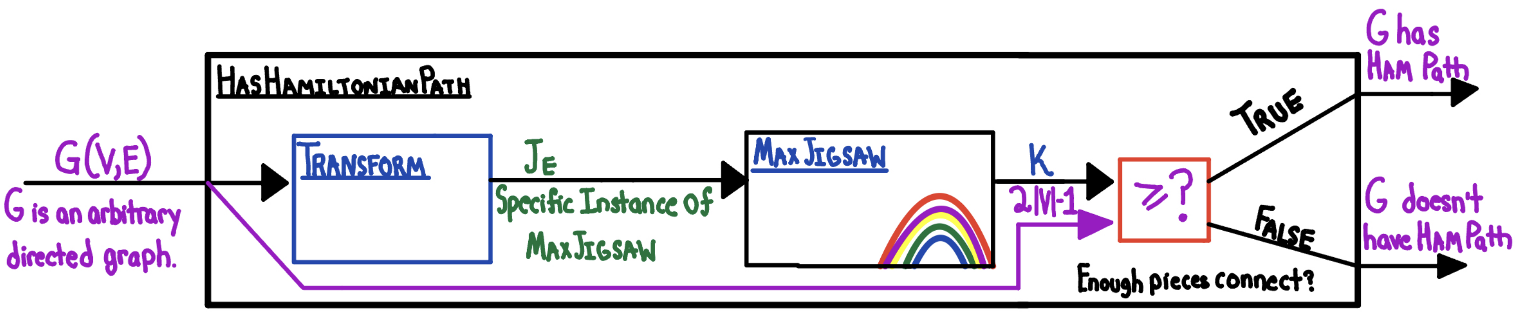

Figure 1: A decider for the directed Hamiltonian Path problem.

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 $\mathbf{G}(\mathbf{V}, \mathbf{E})$ is transformed using a polynomial time reduction, Transform, into a specific instance of the **MaxJigsaw** problem $\mathbf{J_e}$. 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` $= [(v_i, v_j) | \exists$a directed edge from v_i 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 $O(|V||E|) + O(|V|^2) = O(|V|^2)$.

Transform(G(V,E))

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

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

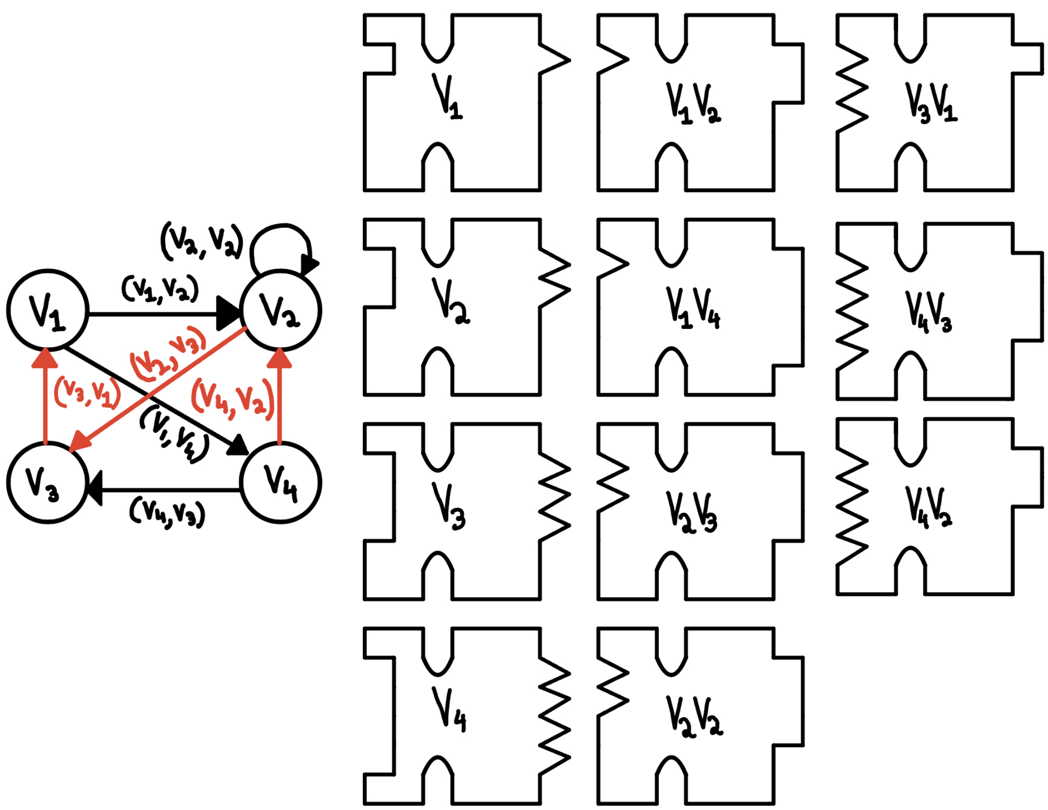

Figure 2: G(V,E) depicts an instance of the Hamiltonian Path problem that is passed as input into Transform(G(V,E)) to produce six jigsaw puzzle pieces $\mathbf{J_e}$, an instance of **MaxJigsaw**.

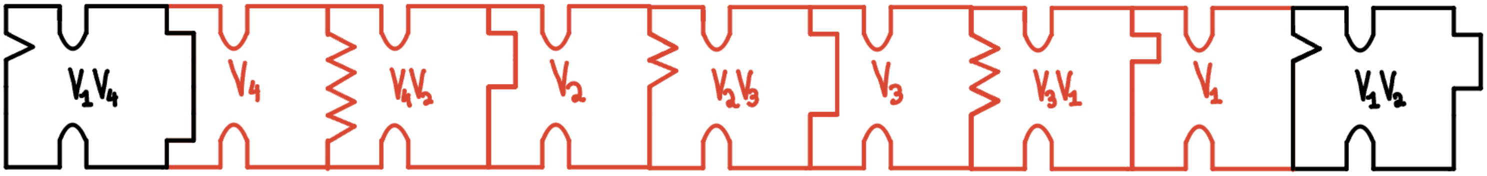

Figure 3: An illustration of the maximum number of connectable jigsaw pieces from Figure 2.

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 $\mathbf{J_e}$ in generated Figure 2, the oracle **MaxJigsaw** determines that a maximum of 9 consecutive puzzle pieces can be connected. Since 9 >= $2|4| - 1$, **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 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: $\mathbf{G(V,E)}$ has a Hamiltonian path $\Rightarrow \geq \mathbf{2|V| - 1}$ pieces in J_e are connected

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

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

$\Rightarrow$ $\forall$ $v_i \in$ V, $\exists$ a corresponding a piece with i left sided square indents i right sided teeth extrusions

$\Rightarrow$ $\forall$ edges (v_i, v_j), $\exists$ a piece with i left sided teeth indents and j right sided square extrusions

$\Rightarrow$ $\forall$ $|V|$ vertices $\in v_{start}$, $e_{(start,i)}$, v_i, $e_{(i,j)}....v_{fin}$, $\exists$ a corresponding piece in J_e and $\forall$ $|V| - 1$ edges $\in$ G connecting two vertices, $\exists$ a puzzle piece connecting the two corresponding vertex pieces

$\Rightarrow \geq 2|V| - 1$ pieces in J_e are connected

.

Sufficiency: $\geq \mathbf{2|V| - 1}$ pieces in J_e are connected $\Rightarrow$ $\mathbf{G(V,E)}$ 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.

$\geq 2|V| - 1$ pieces in J_e are connected

$\Rightarrow$ all vertices of V are included somewhere in the sequence (due to geometric constraints)

$\Rightarrow$ there are $|V| - 1$ edge pieces included in the sequence between the $|V|$ pieces

$\Rightarrow$ $\exists$ an alternating sequence of vertex and edge pieces of size the $2|V| - 1$ including all vertex pieces

$\Rightarrow$ $\exists$ a corresponding alternating sequence of vertices and edges in G that visits every vertex once

$\Rightarrow$ G has a Hamiltonian path

2.3 MaxJigsaw $\in$ co-NP

A NO instance **I** of **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 $\notin$ NP-complete

Since **MaxJigsaw** $\notin$ 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

[DD02] Erik Demaine and Martin Demaine. "Jigsaw Puzzles, Edge Matching, and Polyomino Packing: Connections and Open Problems". In: *Graphs and Combinatorics* 18.1 (2002), pp. 19–36.

[KJS17] Yoon Kim, Yacine Jernite, and David Sontag. "Deep Learning for NLP". In: *arXiv preprint arXiv:1701.00146* (2017).

[Eri19] Jeff Erickson. *Algorithms*. Champaign, IL: University of Illinois, 2019.

[Eri19] Diagram Style
[DD02] Exploring Problem Formulations
[KJS17] Sample Reductions