SHOTGUN ASSEMBLY OF RANDOM JIGSAW PUZZLES

Charles Bordenave* Uriel Feige† Elchanan Mossel‡

October 1, 2019

Abstract

We consider the shotgun assembly problem for a random jigsaw puzzle, introduced by Mossel and Ross (2015). Their model consists of a puzzle – an $n \times n$ grid, where each vertex is viewed as a center of a piece. Each of the four edges adjacent to a vertex is assigned one of q colors (corresponding to "jigs", or cut shapes) uniformly at random. *Unique assembly* refers to there being only one puzzle (the original one) that is consistent with the collection of individual pieces. We show that for any $\varepsilon > 0$, if $q \geq n^{1+\varepsilon}$, then unique assembly holds with high probability. The proof uses an algorithm that assembles the puzzle in time $n^{\Theta(1/\varepsilon)}$.

1 Introduction

[Mossel and Ross, 2015] recently suggested the following problem: Consider a factory that manufactures jigsaw puzzles. The factory aims to make sure that a unique assembly of the puzzle is guaranteed just from the way the pieces are cut, regardless of whether the images on the pieces of the puzzle are informative. In particular, it might be that the factory is producing so called *blank jigsaw puzzles* (which the buyers decorate by themselves), or so called *white jigsaw puzzles* that certain puzzle enthusiasts enjoy assembling.

Suppose that there are q different type of jigs (cut shapes between adjacent pieces), that the puzzle is of size $n \times n$, and that the type of jig between any two adjacent pieces is selected at random. We are interested in the following question:

Question 1.1. *How large should q be so that a random puzzle drawn from this distribution has unique assembly?*

This problem, which [Mossel and Ross, 2015] called the "shotgun assembly of random jigsaw puzzle", is a two dimensional variant of the well studied problem of shotgun assembly of DNA sequences, which is extensively studied from both the combinatorial and probabilistic view points, see for example [Arratia et al., 1996], [Dyer et al., 1994], and [Motahari et al., 2013].

Let us present the above question in a formal manner where the puzzle will be defined as the n by n grid graph with a uniform q coloring of the edges of the grid. From now on we will use the graph theoretic notion of color instead of jig (cut shape, also referred to as "knobs","locks","tabs", "slots", "indents" etc. in the jigsaw puzzle terminology). The parameters for our model are two positive integers, n and q. We use the notation $[m]$ to denote the set of numbers $\{1, \dots, m\}$, and $[a, b]$ to denote the set $\{a, a+1, \dots, b-1, b\}$. A puzzle may be thought of as an n by n grid with colored edges. The building blocks of the puzzle are *pieces* - i.e., vertices of the grid along with 4 adjacent colored half edges. Observe that every vertex

[*] CNRS and Université Aix Marseille; charles.bordenave@univ-amu.fr Research supported by Grants ANR-16-CE40-0024-01 and ANR-14-CE25-0014.

[†] Weizmann Institute; uriel.feige@weizmann.ac.il. Research supported in part by the Israel Science Foundation (grant No. 1388/16). This work was partly done while visiting Microsoft Research, Redmond, Washington.

[‡] MIT; elmos@mit.edu Research supported by NSF grant CCF-1320105 and DMS-1737944 , DOD ONR grants N00014-14-1-0823 and N00014-17-1-2598, and Simons Foundation grant 328025 and Investigator award 622132

not on the boundary of the grid is incident with exactly 4 edges. We assume for simplicity of the presentation (this will not significantly effect the results in the current manuscript) that also every vertex on the boundary is incident with 4 edges. This involves introducing boundary edges that lead out of the grid and do not have vertices at their other endpoint. We further assume for simplicity that at any given vertex v, the edges incident with it are labeled by their orientation: Up, Down, Right and Left and denoted $\uparrow(v)$, $\downarrow(v)$, $\rightarrow(v)$ and $\leftarrow(v)$. We denote by σ the coloring, so that the colors incident to v are $\sigma(\uparrow(v))$, $\sigma(\downarrow(v))$, $\sigma(\rightarrow(v))$ and $\sigma(\leftarrow(v))$. Each edge (including the boundary edges) is given a random color in $[q]$ (corresponding to the type of jig being used), uniformly at random and independently across edges. Thereafter, the puzzle is disassembled, and its pieces are presented at a random order. At this point, the input is n^2 pieces, where each piece is a vertex with 4 incident edges labeled as Up, Down, Right and Left, and colored by colors from $[q]$. An *assembly* of the pieces is a placement of the vertices on an n by n grid, where for each vertex the edges are oriented in the direction of their labels. The assembly is *feasible* if for every two adjacent vertices the colors that they have for their common edge are the same. We refer to the assembly that gives back the original puzzle as the *planted assembly.*

We say that a puzzle has *unique vertex assembly* if it has only one feasible assembly, namely, the planted assembly. We say that a puzzle has *unique edge assembly* if for every feasible assembly and for every edge location (not including boundary edges), the color of the respective edge is the same as in the planted assembly. Note that a puzzle with two identical pieces will not have unique vertex assembly, but it may have unique edge assembly.

Since the probability of having each type of piece is q^{-4}, it follows by the birthday paradox that two identical pieces exist with high probability as soon as $q = o(n)$, and in this case the puzzle does not have unique vertex assembly. It is further shown in [Mossel and Ross, 2015] that if $q = o(n^{2/3})$ then with high probability a random puzzle will not have unique edge assembly. [Mossel and Ross, 2015] further provided a linear time algorithm for unique vertex assembly when $q \geq Cn^2$ for a sufficiently large constant C.

A main open problem from [Mossel and Ross, 2015] is to obtain more accurate bounds for the jigsaw assembly problem. Here we improve the upper bound by proving the following:

Theorem 1.2. *For every $\varepsilon > 0$, if $q \geq n^{1+\varepsilon}$ then with high probability a random puzzle has unique vertex assembly. Moreover, there is an algorithm running in time $n^{O(1/\varepsilon)}$ that with high probability finds the planted assembly.*

Here and elsewhere, the expression "with high probability" means with probability going to 1 as $n \to \infty$.

1.1 Overview of the proof

The proof of Theorem 1.2 is based on the following principle. For a given integer parameter $k > 1$ (where k is a constant independent of n), we refer to a $2k+1$ by $2k+1$ grid as a *window*, and index it by $[-k, k] \times [-k, k]$. Given an input of n^2 pieces, for each piece v, we consider all possible sets of $(2k+1)^2$ pieces (including v itself) and check if they can be assembled as a feasible (namely, legally colored) window with v at its center. A feasible assembly of a window with v at its center will be referred to as a v-window. Given a v-window, the neighborhood $\{(0, \pm 1), (\pm 1, 0)\}$ of v in the v-window is considered to be a *candidate neighborhood* (or in more detail, an ℓ_1 radius 1 candidate neighborhood) of v in the puzzle.

If a piece v is at distance at least $k+1$ from the boundary of the puzzle, the planted assembly will provide a v-window. However, for every constant k, it is likely that there will be pieces v that have additional v-windows, beyond the one provided by the planted assembly. The main insight in our work is that for a sufficiently large constant k (specifically, a choice of $k > 1/\varepsilon$), with high probability, for every piece at distance at least $k+1$ from the boundary of the puzzle, its ℓ_1 radius 1 candidate neighborhood is unique. Namely, the piece v may have several different v-windows, but all of them induce the same candidate neighborhood.

Having established the uniqueness of candidate neighborhoods, this rigidity allows us to assemble the part of the puzzle at distance $k+1$ from the boundaries of the puzzle. A simple algorithm then allows to assemble the rest of the puzzle.

We now provide more details on some aspects of the above proof approach. Consider the planted assembly of the whole puzzle. We shall use the term *region* to denote a set of pieces that forms a connected component in the grid graph of the planted assembly. The region can have any size between 1 (a single piece) and n^2 (the whole grid). We are interested in regions that can be bounded inside an $2k+1$ by $2k+1$ box, as these are the regions that may fit into a v-window. We refer to them as *feasible regions.* The total number of feasible regions in the n by n grid is $C(k)n^2$, where $C(k)$ is a constant that depends only on k. This is because there are n^2 choices for the location ℓ of one piece in the feasible region, and thereafter the remaining pieces of the region need to be chosen from a set of $O(k^2)$ pieces that are at distance at most $O(k)$ from ℓ.

A piece v at distance more than k from the boundary of the puzzle has a *canonical v-window*, which is the feasible region of size $2k+1$ by $2k+1$ that has v at its center. However, if k is too small, then v will have many other v-windows that are unrelated to the canonical one (except for the fact that v is at the center of the v-window). For example, one may create a window by placing in it $(2k+1)^2$ random pieces. There are roughly $n^{2(2k+1)^2}$ ways of choosing these pieces. For the window to be feasible (namely, to be v-window for the piece v that happens to be at its center), these pieces need to satisfy $4k(2k+1)$ constraints, namely, that for every two adjacent pieces in the window, both sides of the edge joining them have the same color. As the coloring σ is random and there are $q = n^{1+\epsilon}$ colors, the probability that all $4k(2k+1)$ constraints are satisfied is $q^{-4k(2k+1)}$. If $q^{-4k(2k+1)}n^{2(2k+1)^2} > 1$ then we expect to have windows that are totally unrelated to canonical windows. To avoid this, we require $q^{2k} > n^{2k+1}$, implying that $k = \Omega(1/\varepsilon)$.

Even if $k > 1/\varepsilon$, there are still likely to be windows that are not canonical (for their respective v). For example, one may try to create a window by taking a $2k+1$ by $2k+1$ feasible region (there are roughly n^2 choices available for this), and replace the piece at its bottom left corner with one of the other pieces of the puzzle (there are roughly n^2 choices for this other piece). The probability for every individual piece to have the colors at its top and right edges be the same as those of the replaced piece is $q^{-2} = n^{-2-2\varepsilon}$. Hence we expect there to be roughly $n^4q^{-2} = n^{2-2\varepsilon}$ such non-canonical windows.

Given that there will be non-canonical windows, what we prove is that with high probability, for every v, for every v-window (whether canonical or not), the candidate neighborhood is the same. To see why such a claim is plausible, consider a v-window that differs from the canonical one by replacing the piece p at location $(0,1)$ by a different piece p' from the puzzle. As location $(0,1)$ is not on the boundary of the window, this requires p' to have all its four edges colored with the same colors as the edges of p, and this event has probability q^{-4}. This probability is low enough to imply that with high probability, there will not be any piece v that has a v-window that differs only in location $(0,1)$ from the canonical v-window.

The above plausibility argument is incomplete, because there may be many ways in which a v-window can be non-canonical, and induce a different candidate neighborhood. At a high level, we address this issue in the following way. The contents of each non-canonical window are composed of feasible regions (where the size of each feasible region is between 1 and $(2k+1)^2-1$) that are placed side by side inside the window. Suppose that the number of feasible regions used is t. On the one hand, the number of possible ways of choosing these feasible regions is at most $(C(k)n^2)^t$. On the other hand, for the window to be feasible, the colors along the boundaries between every two feasible regions (that are placed side by side in the window) need to match. If the total length of these boundaries can be shown to be higher than $2t$, then the probability that the colors match would be smaller than q^{-2t}, and a union bound would show that the non-canonical window is unlikely to exist. Hence it appears that what we need to prove is that if $(0,0)$ and (say) $(0,1)$ come from different feasible regions (this represents a case in which the candidate neighborhood differs from the canonical candidate neighborhood), then indeed the total length of the boundaries is larger than $2t$.

However, the situation is more complicated than that described above. A problem that remains is that not all constraints are independent. Such an issue is illustrated in Example 3.4 and the accompanying Figure 1. For this reason, in our proof, we will not view the window as composed of feasible regions, but rather as composed of *tiles*. A tile is a region of the n by n grid graph whose pieces appear in the window, but the pieces do not have to obey in the window the same adjacency relations that they obey in their origin region. In particular, though the number of pieces in a tile cannot exceed $(2k+1)^2$, the tile need not be enclosed in a $2k+1$ by $2k+1$ box in the n by n grid (though its pieces do need to stay connected). Our main lemma, Lemma 3.6, shows that for a v-window that induces a non-canonical candidate neighborhood, the number of independent constraints is at least roughly twice as large as the number of tiles. The proof of this lemma makes use of isoperimetric inequalities for the grid. Armed with Lemma 3.6, we prove Theorem 2.1 that shows that with high probability, all v-windows induce the same candidate neighborhood.

1.2 Organization of this paper

The paper is organized as follows. In Section 2, we present formal definitions, and statement of the main results on the v-window. In Section 3, we formulate the problem of feasibility of an assembly in graph theoretic terms. Section 4 contains the isoperimetric analysis that is used in our proof. Section 5 describes the reconstruction algorithm.

The appendix (Section A) discusses natural ways by which the assembly model can be modified. One such modification is that pieces need not be given with their orientation – they can be rotated. Another modification is that jigs may have shapes instead of colors, and then two pieces can be placed side by side if their jigs have complementary shapes (rather than the same shape). Theorem 1.2 extends also to these modified setting.

1.3 Concurrent and Follow Up Work

During the preparation of an earlier version of our manuscript [Bordenave et al., 2016], we learned of an independent work by Nenadov, Pfister and Steger which proves similar results. In particular they show like we do that if $q \geq n^{1+\varepsilon}$ for any $\varepsilon > 0$ then there is unique assembly. The model of [Nenadov et. al] assumes that the puzzle has flat boundaries – this makes the problem a little easier as it allows to identify corner pieces and boundary pieces (see Section A for discussion of variants of the model). We do not know if this is necessary for their proof. The proof of [Nenadov et. al] does not provide an (efficient) algorithm for assembly while ours does.

The simple fact that unique vertex assembly does not hold when $q = o(n)$ is implicit in [Mossel and Ross, 2015] – indeed a more elaborate but similar argument in [Mossel and Ross, 2015] shows that if $q = o(n^{2/3})$ then unique edge assembly does not hold. A recent and much more interesting entropy based argument by Martinsson [Martinsson, 2016] shows that if $q < (2/e^{1/2} - \varepsilon)n$, then unique edge assembly is not possible. Combined with the constructive results, this shows that $q = n^{1+o(1)}$ is the correct threshold for both unique vertex reconstruction and unique edge reconstruction.

Finally in two independent recent works [Balister et al., 2017, Martinsson, 2017], it is shown that if $q \geq (2+\varepsilon)n$ then the puzzle has unique edge assembly and if $q \geq \omega(n)$, it has unique vertex assembly. The proofs of [Balister et al., 2017, Martinsson, 2017] builds on the ideas presented in the current paper. In particular the sharper results are obtained by a better analysis of $k \times k$ windows. However, since the results of [Balister et al., 2017, Martinsson, 2017] involve polynomially large windows they do not provide a polynomial time algorithm for assembly.

2 Local Assembly

In our work, we encounter graphs that are grid graphs (and also a graph that is referred to as the *constraint graph*, which is not a grid, and will be introduced in Section 3). Vertices of grid graphs are pieces of the puzzle (and will be referred to as pieces rather than vertices), and two pieces in a grid graph are neighbors if they are end points of the same grid edge. One grid graph is the planted assembly, giving an n by n grid. Every v-window referred to in the introduction is another grid graph, of size $2k+1$ by $2k+1$. Locations of pieces in grid graphs are given as pairs (i,j), where i is the column number (increasing from left to right) and j is the row number (increasing from bottom up).

We now present terminology and notation that is associated with v-windows. For a piece $v \in [n]^2$, let $S_k(v)$ denote the set of injective maps $f : [-k,k]^2 \to [n]^2$ such that

- $f(0,0) = v$ and
- $(f(i,j) : -k \leq i \leq k, -k \leq j \leq k)$ is *feasible*, that is $\sigma(\rightarrow (f(i,j))) = \sigma(\leftarrow (f(i+1,j)))$ for all i,j s.t. $(i,j),(i+1,j) \in [-k,k]^2$, and $\sigma(\uparrow (f(i,j))) = \sigma(\downarrow (f(i,j+1)))$ for all i,j s.t. $(i,j),(i,j+1) \in [-k,k]^2$

Note that $S_k(v)$ may be empty if v is of distance less than k from the boundaries of the grid. Otherwise, $S_k(v)$ contains at least one element, namely the one given by $f(x) = v + x$ for all $x \in [-k,k]^2$. Let $S'_k(v)$ denote the subset of $f \in S_k(v)$ where there exists an $\alpha \in \{\pm(0,1), \pm(1,0)\}$ with $f(\alpha) \neq v + \alpha$. Hence, $S_k(v) \backslash S'_k(v)$ is the subset of local assembly whose restriction to the candidate neighborhood $\{(0,0), \pm(0,1), \pm(1,0)\}$ coincides with the the planted assembly.

We use the notation $C(k)$ to denote a constant that only depends on k. Different occurrences of $C(k)$ may refer to different constants, though sometimes for clarity, we will also use the notation $C'(k)$ so as to indicate that the constant is different from another constant $C(k)$.

The main theorem we wish to prove is the following:

Theorem 2.1. *There exists $c > 0$ such that for all $\varepsilon > 0$, if $k \geq c/\varepsilon$ and $q \geq n^{1+\varepsilon}$ then the following holds: For every $v \in [n]^2$,*

$$P[S'_k(v) \text{ non-empty }] \leq C(k) n^{-2-\varepsilon/2}.$$

Theorem 2.1 is the main result needed to prove that with high probability all vertices at distance at most k from the boundaries can be assembled correctly. A simple algorithm then allows to construct the reminder of the puzzle. This will allow us to establish Theorem 1.2.

3 The Constraint Graph

The proof of Theorem 2.1 is based on a detailed analysis of the constraints imposed by the condition that an injective function $f : [-k,k]^2 \to [n]^2$ is feasible, along with isoperimetric reasoning in order to lower bound the number of independent constraints. In this section, we formalize these notions and state a key lemma, Lemma 3.6. Using this lemma (whose proof appears in Section 4) we prove Theorem 2.1.

To simplify notation we write $(i, j : j+1)$ for the edge $((i,j),(i,j+1))$. Similarly we write $(i : i+1, j) := ((i,j),(i+1,j))$. Note that by definition

$$\rightarrow (i,j) = (i : i+1, j) = \leftarrow (i+1,j), \quad \uparrow (i,j) = (i, j : j+1) = \downarrow (i, j+1).$$

Sometimes it would be more useful to analyze the constraints imposed by f on a subset of $[-k,k]^2$. This leads to the following definitions:

Definition 3.1. For a given $f:[-k,k]^2 \to [n]^2$, and $W \subset [-k,k]^2$, the *restriction of f to W*, denoted $f_{|W}$, is the function $f_{|W}: W \to [n]^2$ defined by $f_{|W}(w) = f(w)$, for all $w \in W$.

Given $f:[-k,k]^2 \to [n]^2$ and $W \subset [-k,k]^2$, the image of $f_{|W}$ is a set of pieces (vertices) in the n by n grid graph. The *tiles* of $f_{|W}$, denoted $T(f_{|W})$, is the collection of connected components formed by this image. Namely, each tile corresponds to a connected component of the graph with vertex set $f(W)$, and where vertices v, w are adjacent if $v-w \in \{\pm(0,1), \pm(1,0)\}$. We write $T(f)$ for $T(f_{|[-k,k]^2})$ and call $T(f)$ the tiles of f.

We now introduce the *constraint graph*, whose vertices correspond to edges of the n by n grid graph, and two vertices of the constraint graph are joined by a constraint graph edge if the corresponding grid-edges need to have the same color (in order for a certain v-window to be feasible).

Definition 3.2. The constraint graph $G(f_{|W}) = (V,E)$ of $f_{|W}$ for $f:[-k,k]^2 \to [n]^2$ is the graph whose edge set E consists of

$$\begin{array}{lll} (\to (f(u)), \leftarrow (f(u+(1,0))) & \text{if } f(u+(1,0)) \neq f(u)+(1,0) & \text{and } u, u+(1,0) \in W, \\ (\uparrow (f(u)), \downarrow (f(u+(0,1))) & \text{if } f(u+(0,1)) \neq f(u)+(0,1) & \text{and } u, u+(0,1) \in W. \end{array}$$

The vertex set $V(f_{|W})$ of $G(f_{|W})$ is the set of all edges of $[n]^2$ spanned by E. An edge of $G(f_{|W})$ will be called a constraint. The constraint graph of f is the constraint graph of $f_{|W}$ for $W = [-k,k]^2$. To distinguish the vertices and edges of the grid from those of G, we will sometimes write explicitly G-vertices and G-edges and grid-vertices (or pieces) and grid-edges. We write $c(f_{|W})$ for the number of connected components of G and $\gamma(f_{|W}) = |V(f_{|W})| - c(f_{|W})$. We will omit the subscript W when $W = [-k,k]^2$.

Observe the following:

- The degree of each G-vertex is at most 2, because a grid-edge (that corresponds to the G-vertex) can be in a constraint based on either end of that grid-edge. Therefore the connected components of $G(f_{|W})$ are either paths or cycles.
- The constraint graph may have pairs of parallel edges. This is shown in Example 3.4 and its accompanying Figure 1. Hence the constraint graph may have cycles of length 2 (formed by a pair of parallel edges).

Consider a candidate $f:[-k,k]^2 \to [n]^2$. We say that an edge $((i:i+1,j),(i':i'+1,j'))$ of the constraint graph $G(f)$ is *satisfied* if $\sigma((i:i+1,j)) = \sigma((i':i'+1,j'))$ and similarly for an edge $((i,j:j+1),(i',j':j'+1))$. We say that $G(f_{|W})$ is *satisfied* if all of its edges are satisfied.

The following lemma implies that the term $\gamma(f_{|W})$ (introduced in Definition 3.2) corresponds to the number of independent constraints that need to be satisfied in order for $f_{|W}$ to be feasible.

Lemma 3.3. *$f_{|W}$ is feasible iff $G(f_{|W})$ is satisfied. Moreover, for a fixed $f:[-k,k]^2 \to [n]$ and $W \subset [-k,k]^2$, the probability that $f_{|W}$ is feasible for a random puzzle is $q^{-\gamma(f_{|W})}$.*

Proof. The first statement follows from the definitions. For the second statement we will compute the probability that $G(f_{|W})$ is satisfied. For $G(f_{|W})$ to be satisfied, it is required that the color of G-vertices of $G(f_{|W})$ (grid-edges) in each connected component are identical. Note that events for different components are independent and the probability that a certain component C has all G-vertices of the same color is q^{-c+1} where c is the number of G-vertices in C. The conclusion follows. □

Example 3.4. Let $W = [1,2] \times [1,2]$ and let $g = f_{|W}$ be defined by

$$g(1,1) = (1,1), \quad g(1,2) = (3,2), \quad g(2,1) = (3,1), \quad g(2,2) = (1,2).$$

In this case, the map $f_{|W}$ has 2 tiles, namely {(1,1),(1,2)}, {(3,1),(3,2)}. The constraint graph is the graph with the following edges:

$$((1,1:2),(3,1:2)), \quad ((1,1:2),(3,1:2)), \quad ((1:2,1),(2:3,1)), \quad ((3:4,2),(0:1,2))$$

Note that the first two edges are parallel to each other, as one is derived from the adjacency of $(1,1)$ below $(3,2)$, and the other from the adjacency of $(3,1)$ below $(1,2)$. The vertex set V of $G(f)$ consists of

$$(1,1:2),(3,1:2),(1:2,1),(2:3,1),(3:4,2),(1:0,2)$$

and is of size 6. Two of the connected components of $G(f_{|W})$ are given by the last two edges, and the remaining connected component is a cycle of length 2 given by the parallel edges. Thus $|V| = 6$, the number of connected components is 3, and $\gamma(f_{|W}) = 6 - 3 = 3$. The probability that $f_{|W}$ is feasible is $q^{-\gamma(f_{|W})} = q^{-3}$. See Figure 1 (right).

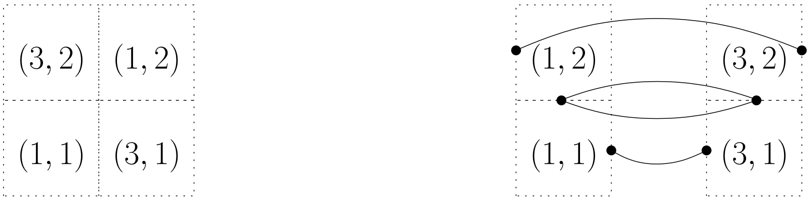

Figure 1: The local assembly of W in Example 3.4 is depicted on the left. The result of the map $g = f_{|W}$ is depicted on the right, together with the six vertices and four edges of its constraint graph. The four edges give rise to only three independent constraints, because two of the edges are parallel to each other (or equivalently, close a cycle).

Lemma 3.5. *Let $u(f_{|W})$ denote the number of edges (constraints) of $G(f_{|W})$ that have as an endpoint a G-vertex of degree 1, and let $w(f_{|W})$ denote the total number of constraints. Then $\gamma(f_{|W}) \geq u(f_{|W}) + 0.5(w(f_{|W}) - u(f_{|W})) = 0.5w(f_{|W}) + 0.5u(f_{|W})$.*

Proof. As noted earlier the degree of each vertex in $G(f_{|W})$ is at most two. Therefore the graph $G(f_{|W})$ is a disjoint union of cycles and paths. Moreover, γ, u and w are all additive over disjoint components. Therefore it suffices to check the claim for each component separately. For a path with one edge we have $\gamma = u = w = 1$ as needed. For a path with $\ell \geq 3$ vertices :

$$\gamma = \ell - 1, u = 2, w = \ell - 1$$

so the inequality holds in this case as well. For a cycle of length $\ell \geq 2$ we have:

$$\gamma = \ell - 1, u = 0, w = \ell$$

and the inequality also holds in this case. □

Recall the definition of $S'_k(v)$ above Theorem 2.1 and the definition of tiles in Definition 3.1. It involves two aspects: feasibility with respect to a coloring σ, and the inequality $f(\alpha) \neq v + \alpha$ for some $\alpha \in \{\pm(0,1), \pm(1,0)\}$. The main result of the next section is Lemma 3.6, which for simplicity of notation is stated for $S'_k(v)$, even though its proof only uses the inequality aspect of $S'_k(v)$ and not its feasibility aspect. For any $f \in S'_k(v)$, Lemma 3.6 provides a lower bound on the parameter γ for a well-chosen W in terms of a parameter which plays the role of the degrees of freedom of $f(W)$ (the integer t in the lemma). Such a lower bound provides an upper bound on the probability that f is feasible, by Lemma 3.3.

Lemma 3.6. *For $f \in S'_k(v)$, let $T(f) = (T_i)_i$ be the collection of tiles in $[n]^2$ determined by f. Then for every $\varepsilon > 0$ if $k > c/\varepsilon$ for a large enough c then the following holds. For every $f \in S'_k(v)$, there exists a W such that $0 \in W \subset [-k,k]^2$ with the following property. Let $t+1 = |\{i : f(W) \cap T_i \neq \emptyset\}|$. Then $(1+\varepsilon)\gamma(f_{|W}) \geq 2t+2+\varepsilon$.*

The proof of Lemma 3.6 involves two aspects. One aspect is isoperimetric results on the grid graph, that can be used in order to lower bound the number of independent constraints (not all constraints are independent, as can be seen in the example appearing in Figure 1) that are associated with each tile. The other aspect is a case analysis that depending on the number and sizes of tiles, selects an appropriate W for which the lemma can be proved. The full proof of Lemma 3.6 appears in Section 4. Here we assume that Lemma 3.6 holds, and now prove Theorem 2.1.

Proof of Theorem 2.1. Recall that $q \geq n^{1+\varepsilon}$. Given a piece $v \in [n]^2$, let B_v denote the bad event that $S'_k(v)$ is non-empty. Namely, B_v is the event that there exists a feasible f, where $f : [-k,k]^2 \to [n]^2$ with $f(0) = v$ and $f(\alpha) \neq v + \alpha$ for some $\alpha \in \{\pm(0,1), \pm(1,0)\}$. B_v is a random event, as it depends on the random coloring σ with q colors of the edges of the n by n grid. We need to upper the probability of B_v happening.

By Lemma 3.6, if B_v happens, then there is some $W \subset [-k,k]^2$ with $0 \in W$, for which $f_{|W}$ is feasible and moreover $(1+\varepsilon)\gamma(f_{|W}) \geq 2t+2+\varepsilon$, where $t+1$ is the number of tiles that $f(W)$ intersects.

The number of choices for W is some constant $C'(k)$ that depends only on k. Given W and the fact that $f(0) = v$, the number of choices of $f_{|W}$ is at most $C(k)n^{2t}$ (where $C(k)$ is larger than $C'(k)$, but it too is a constant that depends only on k). This follows because each tile T is determined by one $f(w) \in T$ (and there are at most n^2 possibilities for $f(w)$), and all its remaining pieces (let $p \leq (2k+1)^2$ denote the number of pieces in the tile) form a connected component in the n by n grid. As the grid has maximum degree 4, the number of connected components of size p that contain a predetermined grid vertex is at most 2^{3p}.

By Lemma 3.3, the probability that $f_{|W}$ is feasible is bounded above by $q^{-\gamma(f_{|W})}$, which can be bounded by $n^{-2t-2-\varepsilon}$ by Lemma 3.6.

It follows that the overall probability that such an f exists with $f(0) = v$ is upper bounded by $C(k)n^{-2-\varepsilon}$ as needed. □

4 Isoperimetric Analysis

In this section, we will prove the main isoperimetric lemma, i.e. Lemma 3.6. For a connected subset T of $[n]^2$ we let ∂T denote the *edge boundary* of T and $|\partial T|$ denote the length of the boundary, i.e., the number of edges between T and its complement. We start by proving the following lemma:

Lemma 4.1. *Let $f \in [-k,k]^2 \to [n]^2$ with the number of tiles in f, $|T(f)| = t+1 \geq 2$. Then*

$$\gamma(f) \geq 2t(1 - \frac{1}{s}),$$

where $s = 2k+1$. Moreover, if there are two tiles that each have more than 35 pieces then

$$\gamma(f) \geq 2t(1 - \frac{1}{s}) + 4.$$

Our proof will be based on the following classical fact.

Lemma 4.2. *Let $A \subset R^2$ be a set with boundary that is axis aligned. Then the length of its boundary ∂A satisfies $|\partial A| \geq 4|A|^{1/2}$, where $|A|$ is the area of the set.*

A special case of the lemma above is the elementary exercise showing that the square minimizes the surface area among all rectangles of a given area. The more general case can be proved for example by looking at the minimal axis align rectangle containing the body A and observing that its surface area must be smaller or equal to the surface area of A. The following elementary arithmetic lemma will be used in the proof of Lemma 4.1.

Lemma 4.3. *If $a_0 \geq a_1 \geq \cdots a_t \geq 1$ is an integer partition of s^2, $\sum_i a_i = s^2$, let*

$$g = 2\sum_{i=0}^{t} \sqrt{a_i} - 2s.$$

Then

$$g \geq 2t(1 - \frac{1}{s}).$$

Moreover if $a_0 \geq a_1 \geq 36$ then

$$g \geq 2t(1 - \frac{1}{s}) + 4.$$

Proof. Since $x \to x^{1/2}$ is concave, the minimum of g under the constraints that $\sum a_i = s^2$ and each $a_i \geq 1$ is obtained when all of the a_i but one, satisfy $a_i = 1$. Thus

$$g \geq 2(t + s\sqrt{1 - t/s^2} - s) \geq 2(t + s(1 - t/s^2) - s) = 2t(1 - \frac{1}{s}).$$

The first statement proof follows. When $a_0 \geq a_1 \geq 36$, utilizing the concavity of $x^{1/2}$ allows to obtain a better bound. consider the integer partition b obtained by joining all the mass of a_1 to a_0 except one unit that is left separately:

$$b_0 = a_0 + a_1 - 1, \quad b_1 = a_2, \ldots, b_{t-1} = a_t, \quad b_t = 1.$$

Since $\sqrt{a_0} \geq \sqrt{a_1} \geq 6$ we get

$$10 + 2\sqrt{a_0}\sqrt{a_1} \geq 2\sqrt{a_0}\sqrt{a_1} \geq 6(\sqrt{a_0} + \sqrt{a_1})$$

This implies

$$(\sqrt{a_0} + \sqrt{a_1} - 3)^2 = a_0 + a_1 + 9 - 6(\sqrt{a_0} + \sqrt{a_1}) + 2\sqrt{a_0}\sqrt{a_1} \geq a_0 + a_1 - 1,$$

so taking square roots we see that

$$\sqrt{a_0} + \sqrt{a_1} \geq \sqrt{a_0 + a_1 - 1} + 3 = \sqrt{b_0} + \sqrt{b_t} + 2.$$

Hence, the first statement of the lemma gives

$$2\sum_{i=0}^{t} \sqrt{a}_i - 2s \geq 2\sum_{i=0}^{t} \sqrt{b}_i - 2s + 4 \geq 2t(1 - \frac{1}{s}) + 4,$$

as needed. □

We can now prove Lemma 4.1

Proof of Lemma 4.1. Note that except for the edges at the boundary of the grid $[-k, k]^2$, every edge at the boundary of one of the tiles $T_0, \ldots, T_t$ is part of a constraint and appears uniquely. Thus by Lemma 3.5 (in fact, here we use only a weaker inequality than that implied by Lemma 3.5 – the full power of that lemma will be used at later points) it follows that

$$\gamma(f) \geq \frac{1}{2}(\sum_{i=0}^{t} |\partial T_i| - 4s) \geq 2\sum_{i=0}^{t} \sqrt{|T_i|} - 2s.$$

where the second inequality follows from Lemma 4.2. The lemma is then a consequence of Lemma 4.3 □

We now prove Lemma 3.6.

Proof of Lemma 3.6. We will take $c = 200$ so $k \geq 200/\varepsilon$. In order to define W, we will consider a few cases on the structure of f. We will consider a few cases. Let $\tau + 1$ be the number of tiles of f.

- $\tau \geq 3/\varepsilon$. In this case, we set $W = [-k, k]^2$. Then $t = \tau$ and Lemma 4.1 imply that

  $$\gamma(f)(1+\varepsilon) \geq 2t(1 - 1/(2k+1))(1+\varepsilon) \geq 2t(1+\varepsilon/2) \geq 2t + 3.$$

  Hence, the set W satisfies the conclusion of Lemma 3.6.

- We next consider the case where the second largest tile is of area at least 36. We may also take $W = [-k, k]^2$. Then $t = \tau$ and by the second part of Lemma 4.1,

  $$\gamma(f)(1+\varepsilon) \geq 2t(1+\varepsilon)(1 - \frac{1}{2k+1}) + 4 \geq 2t + 4,$$

  as needed.

- We next consider the case where $f([-2, 2]^2)$ is all part of the same tile of f. In this case, we set $W = [-2, 2]^2$. Then, we have $t = 0$. Thus it is sufficient to check that $\gamma(f_{|W|}) \geq 2$. To this end, consider the graph H with vertex set W obtained by joining, for $\beta \in \{\pm(1, 0), \pm(0, 1)\}$, x and $x + \beta$ if $f(x + \beta) = f(x) + \beta$. In words, the edges of H correspond to pairs of vertices that are adjacent (and in the same orientation) both in W and in the original puzzle. Therefore, if x and y are in the same connected component of H then $f(y) = f(x + (y - x)) = f(x) + (y - x)$ (this can be proven by induction on the length of the minimal path connecting x and y in H). Hence, our assumption $f(0) = v$ and $f(\alpha) \neq v + \alpha$ for some $\alpha \in \{\pm(1, 0), \pm(0, 1)\}$, implies that 0 and α are not in the same connected component of H. On the other hand, we observe that except for the edges in ∂W, every edge at the boundary of a connected component of H is part of a constraint in $G(f_{|W})$. By inspecting the possible configurations of the connected component of α in G, we see that has a at least 4 edges on its boundary which are not in ∂W. It follows there are at least 4 constraints. By Lemma 3.5, it implies that $\gamma(f_{|W}) \geq 2$ as needed.

- The last case is where $\tau < 3/\varepsilon$, all the parts but one are of area less than 36 and there exist $y, x \in [-2, 2]^2$ which belong to different tiles. Let T_0 be the tile of f with the maximal size. Note that

  $$|T_0| \geq (2k+1)^2 - 3/\varepsilon \times 36 > (2k+1) \times (2k).$$

  Since $f(x)$ and $f(y)$ lie in different tiles, at least one of the two doesn't belong to T_0. Without loss of generality assume that $x \in f^{-1}(T_1)$ where $|T_1| < 36$. Let W' denote the connected component of x in the subset $[-k, k]^2 \setminus f^{-1}(T_0)$. A key observation is that since $\tau \times 36 + 2 < k = 200/\varepsilon$, it follows that none of the elements of W' are adjacent to the boundary of the grid $[-k, k]^2$. In other words each edge in $\partial W'$ has one of its end point in $f^{-1}(T_0)$. This implies that $\partial_v W' \subset f^{-1}(T_0)$, where $\partial_v W'$ is the *vertex boundary* of W'. We set $W = W' \cup \partial_v W'$.

  Define $U_0 = f^{-1}(T_0)$ and let $U_i = f^{-1}(T_i) \cap W$. We assume without loss of generality that $U_i \neq \emptyset$ for $i = 0, \ldots, t$ and U_i is empty otherwise. In other words, $f(W)$ intersects $t + 1$ tiles of f. We wish to lower bound $\gamma(f_{|W})$. Note that every edge between different U_i's defines a constraint. Thus

  $$w(f_{|W}) \geq \frac{1}{2} \sum_{i=1}^{t} |\partial U_i| + \frac{1}{2} |\partial W'|.$$

Moreover, every edge in $\partial f(U_i)$ defines a constraint with a vertex that appears only once. Thus

$$u(f_{|W}) \geq \frac{1}{2}\sum_{i=1}^{t} |\partial f(U_i)|.$$

Thus by Lemma 3.5 and the fact that the boundary of each set is at least 4 it follows that

$$\gamma(f_{|W}) \geq \frac{1}{2}(w(f) + u(f)) \geq \frac{1}{4}\left(\sum_{i=1}^{t} |\partial U_i| + \sum_{i=1}^{t} |\partial f(U_i)| + |\partial W'|\right) \geq 2t + \frac{1}{4}|\partial W'|.$$

If $|W'| \geq 2$ then $|\partial W'| \geq 6$ and so $\gamma(f_{|W}) \geq 2t + 1.5$. However since $\gamma(f_{|W})$ is integer we get $\gamma(f_{|W}) \geq 2t + 2$ and therefore

$$\gamma(f_{|W})(1+\varepsilon) \geq 2t + 2 + \varepsilon,$$

as needed. So it remains to prove the claim when $|W'| = 1$. In this case, $\gamma(f_{|W}) = 4$ and $(1+\varepsilon)\gamma(f_{|W}) \geq 4 + \varepsilon$ as needed.

The proof is complete. □

5 Algorithmic aspects

We now prove our main result Theorem 1.2. This involves two aspects: proving that with high probability there is a unique vertex assembly (Theorem 5.2 below), and bounding the running time of an algorithm that finds this vertex assembly (Theorem 5.3 below). Both aspects follow relatively easily from Theorem 2.1. By constructing all possible v-windows and deriving from them candidate neighborhoods, it is not difficult to assemble together all those pieces whose distance from the boundary of the n by n puzzle is at most k. Thereafter, only $O(nk)$ pieces remain unassembled, which is smaller than the number of colors q. Having more colors than pieces, with high probability there will be many pairs of pieces for which one can determine unambiguously that they must be neighbors. This leads to a greedy algorithm for extending the assembled region to cover the whole puzzle.

The deterministic algorithm that we present uses dynamic programming rather than exhaustive search in order to find all v-windows. Consequently we can bound its running time by $C(k)n^{O(k)}$ (for $k = O(\frac{1}{\epsilon})$), which is better than a running time of $C(k)n^{O(k^2)}$ that would follow from the use of exhaustive search.

We now proceed to prove Theorem 1.2. Throughout this section, we take $k = \lceil c/\varepsilon \rceil$, where $c = 200$ is as in Theorem 2.1 and n large enough so that $2(n-2k)^2 \geq n^2$.

Consider the original planted assembly of the puzzle. In this assembly, we refer to pieces located in $[k+1, n-k] \times [k+1, n-k]$ as *core* pieces, and to other pieces as *peripheral* pieces. We further partition the periphery into k concentric *shells*, where shell k contains those pieces on the boundary of the puzzle, and shell i for $1 \leq i \leq k-1$ containing those pieces at distance $k-i$ from shell k. Shell 0 is defined similarly, it is the inner boundary of the core. An edge is a peripheral edge if it is adjacent to a peripheral piece. A jig of a piece refers to an edge adjacent to a piece.

Recall that for any piece v, for every f in $S_k(v)$ and for $\alpha \in \{(0, \pm 1), (\pm 1, 0)\}$, the set $f(\alpha)$ of four pieces is called a candidate neighborhood of v. Let $c' > 0$ be a fixed constant. We say that a puzzle is *typical* if the following properties hold,

(i) Every core piece v has a unique candidate neighborhood.

(ii) Every peripheral piece v either has no candidate neighborhood or a unique candidate neighborhood. In this last case, this candidate neighborhood is the neighborhood of the piece in the planted assembly.

(iii) The number of peripheral edges with a non-unique color among the peripheral edges is at most $n-2k-1$.

(iv) For every peripheral piece v and two jigs of v (say j_1 and j_2), no other peripheral piece u has two jigs (say j_3 and j_4) with matching colors. Namely, $\sigma(j_1)=\sigma(j_3)$ and $\sigma(j_2)=\sigma(j_4)$ cannot hold simultaneously.

(v) For every two colors a, b there are at most $c'k$ pieces (not necessarily peripheral) with two jigs with these colors.

For a random puzzle, properties (i)-(ii) hold with high probability thanks to Theorem 2.1. The other properties are easy to check. The proof of the following lemma is straightforward and given in the appendix.

Lemma 5.1. *If k is as above and $c' = 4/c = 1/50$ in property* (v)*, with high probability, a random puzzle is typical.*

We now describe a deterministic algorithm that will reconstruct the planted assembly whenever the underlying puzzle is typical. We describe successively each step of the algorithm on a general puzzle and explain how it proceeds on a typical puzzle. We will later explain how to implement it.

1. For each puzzle piece v, determine whether it has a candidate neighborhood. If there is no candidate neighborhood mark the piece v as peripheral. If there is a unique candidate neighborhood note which pieces are the neighbors of v. Finally, if there is a piece with a non-unique candidate neighborhood, the algorithm stops here and fails to reconstruct the planted assembly.

The properties (i)-(ii) imply that the algorithm will not stop for a typical puzzle. Observe also that property (i) implies that all pieces marked as peripheral are indeed peripheral pieces. Note however, that for the other pieces, we do not yet know whether they are peripheral or belong to the core.

2. Greedily join pairs of pieces that are neighbors of each other, as long as possible. If the largest connected component does not contain a $n-2k$ by $n-2k$ square, the algorithm stops and fails.

For a typical puzzle, property (i) implies that all core pieces will belong to the same connected component. The condition $2(n-2k)^2 \geq n^2$ implies that the largest connected component does necessarily contain the core. Hence the algorithm will not stop here. Importantly, properties (i)-(ii) imply that the pieces are necessarily assembled as in the planted assembly.

As noted above, the largest connected component contains the core. In the next step of the algorithm we shall expand the largest connected component until it contains the whole puzzle. For simplicity of the presentation, we first describe the next step of the algorithm (step 3) as if the largest connected component contains exactly the core. Later we explain how to adapt step 3 to the case that the largest connected component also contains pieces not from the core.

3. Greedily assemble the shells of the periphery one by one, from the core towards the boundary of the puzzle as follows. Shell 0 is already assembled. For $0 \leq i \leq k-1$, suppose that shell i was already assembled. To assemble shell $i+1$ find in each one of the four sides of shell i

one piece whose free edge (leading out of the assembled part) has a color that appears only once among the yet unassembled peripheral pieces. (If the puzzle is typical, each side will contain such an edge.) Find the unique yet unassembled peripheral piece that has an edge of the desired color and insert it in its location. Thereafter, the rest of shell $i+1$ is greedily assembled as follows. Consider an undetermined location next to an already assembled piece of shell $i+1$ which is not one of the four corners of shell $i+1$. This undetermined location is a neigbhor of two already assembled pieces, thus it specifies two free edges. Among the yet unassembled pieces, insert here the unique piece which has matching colors with these two free edges. (If the puzzle is typical, there will be a unique such piece.) When, all but the four corners of shell $i+1$ are assembled, the above procedure is applied to the four corners.

Assume that the puzzle is typical and that the largest connected component of step 2 contains exactly the core. To see that step 3 finds the planted assembly, we prove by induction on i, $0 \leq i \leq k-1$, that the algorithm reconstructs correctly shell $i+1$. To this end, notice that property (iii) implies that for each side of shell i, $0 \leq i \leq k-1$, there will be at least one free edge among the $n-2(k-i)$ free edges with a color which appears once among the yet unassembled pieces. Then, thanks to property (iv), we will reconstruct unambiguously shell $i+1$.

Assume now that the puzzle is typical but that the largest connected component of step 2 contains not only the core, but also additional pieces. The adaptation of step 3 to this case contains two aspects. One is that some operations described in step 3 might become redundant (because the respective pieces are already assembled), and hence are not executed. The other is that rather than starting from the core and assembling the periphery one shell at a time, we shall start with the largest enclosed rectangle (which contains the core, but might contain also parts of the periphery, and might not be centered at the core), and extend it by one row or one column at a time (in whatever direction that works). As without the adaptation, this version of step 3 will recover the planted puzzle (if the puzzle is typical). However, due to the adaptation, we need to argue that it will not find also some other legal assembly in addition to the planted assembly, by of (for example) not assembling the bottom row and instead adding a row extending out of the boundary of the puzzle above the top row. Such a possibility is excluded by property (iii): at least one of the free edges on each side of the boundary has a color which is not present among the yet unassembled pieces.

The above analysis of the algorithm has proved its correctness on typical puzzles. (Note that we have not used so far the property (v).)

Theorem 5.2. *If the puzzle is typical then the above algorithm recovers the planted puzzle.*

We now analyze the complexity of the algorithm. Here property (v) will be used.

Theorem 5.3. *If the puzzle is typical then the above algorithm can be implemented to run in time $O(k)^{k^2} n^{O(k)}$.*

Proof. There are at most $\min(q, 4n^2)$ distinct colors in the puzzle. Hence, there are most $n^{O(1)}$ pairs of colors used in the puzzle. In time $n^{O(1)}$, we can build a table which to any such pair of colors returns the set of pieces which have jigs with these two colors. Property (v) implies that for every pair of colors the respective set has cardinality at most $m = c'k = O(k)$.

We perform step 1 of the algorithm by listing all the feasible assemblies of $[-k,k]^2$. This list can be computed in time $O(k)^{k^2} n^{O(k)}$ in the following manner: (a) Enumerate over all possible pieces in the top row and left column of the square. That is, we enumerate all local assembly on $W = (\{-k\} \times [-k,k]) \cup ([-k,k] \times \{k\})$. (b) For each feasible local assembly on W, we enumerate all pieces that can be placed on the top and left corner of $[-k,k]^2 \backslash W$. It gives the set of feasible assembly on $W' = W \cup \{(-k+1, k-1)\}$. (c) We repeat the previous step to W' and proceed sequentially from top to bottom and left to right.

The output of the algorithm is the enumeration of all feasible assembly of $[-k, k]^2$. By exhaustive search, the running time of part (a) is $(n^2)^{2k} = n^{O(k)}$. For the part (b)-(c), the running time to enumerate all feasible assembly whose restriction to W is fixed is $O(m^{k^2}) = O(k)^{k^2}$ where m is as above. It corresponds to the calls in the table which to any pair of colors return the set pieces which have matching colors. Indeed, once the top row and left column are fixed, each new piece has two colors constrained. There are at most $m^{(k-1)^2}$ calls in this table.

In the process of computing this list of all feasible assemblies, when a new feasible assembly on $[-k, k]^2$ is found, we update in time $O(1)$, the candidate neighborhood of the central piece. It follows that step 1 of the algorithm can be performed in time $O(k)^{k^2} n^{O(k)}$.

Step 2 is performed in time $O(n^2)$ by a greedy exploration. In step 3, to reconstruct shell $i + 1$, it first uses time $O(kn^2)$ to find on each side a free edge with a unique color. Thereafter, the reconstruction of the shell can be done in time $O(n)$, using $4(n - 2k - 2i)$ calls to the table referred to above (which for any pair of colors lists the pieces that have jigs with these colors). We obtain the claimed running time for the algorithm. □

References

[Arratia et al., 1996] Arratia, R., Martin, D., Reinert, G., and Waterman, M. S. (1996). Poisson process approximation for sequence repeats, and sequencing by hybridization. *J. Comp. Bio.*, 3(3):425–463.

[Bordenave et al., 2016] C. Bordenave, U. Feige and E. Mossel (2016) Shotgun Assembly of Random Jigsaw Puzzles. arXiv preprint arXiv:1605.03086.

[Balister et al., 2017] P. Balister, B. Bollobás and B. Narayanan (2017) Reconstructing random jigsaws. arXiv preprint arXiv:1707.04730

[Dyer et al., 1994] Dyer, M., Frieze, A., and Suen, S. (1994). The probability of unique solutions of sequencing by hybridization. *J. Comp. Bio.*, 1(2):105–110.

[Grimmett, 1999] Grimmett, G. (1999). *Percolation*, volume 321 of *Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]*. Springer-Verlag, Berlin, second edition.

[Harary, 1974] Harary, F. (1974). A survey of the reconstruction conjecture. In *Graphs and combinatorics (Proc. Capital Conf., George Washington Univ., Washington, D.C., 1973)*, pages 18–28. Lecture Notes in Math., Vol, 406. Springer, Berlin.

[Kelly, 1957] Kelly, P. J. (1957). A congruence theorem for trees. *Pacific J. Math.*, 7:961–968.

[Martinsson, 2016] A. Martinsson (2016). Shotgun edge assembly of random jigsaw puzzles. arXiv preprint arXiv:1605.07151.

[Martinsson, 2017] A. Martinsson (2017). A linear threshold for uniqueness of solutions to random jigsaw puzzles. arXiv preprint arXiv:1701.04813.

[Motahari et al., 2013] Motahari, A. S., Bresler, G., and Tse, D. N. (2013). Information theory of DNA shotgun sequencing. *Information Theory, IEEE Transactions on*, 59(10):6273–6289.

[Mossel and Ross, 2015] Mossel, E. and Ross, N (2015). Shotgun assembly of labeled graphs. Arxiv preprint 1504.07682.

[Nenadov et. al] R. Nenadov, P. Pfiser and A. Steger (2016). Unique reconstruction threshold for random jigsaw puzzles. arXiv preprint arXiv:1605.03043.

A Variants

The model that we have studied can be generalized to a model where the jigs have a shape and the pieces are allowed to be rotated. This could be formalized using (oriented) edges as follows. The set of edges $e = (x, y)$ of the grid such that $x \in [n]^2$ is denoted by E. The set E^{in} is the subset of edges such that both x and y are in $[n]^2$. It is stable under the involution $\check{\cdot}$ defined for every $e = (x, y)$ by $\check{e} = (y, x)$. The edges adjacent to $x \in \mathbf{Z}^2$ are organized in counter-clockwise order (right, up, left and down), we set

$$x + B = ((x, x + (1, 0)), (x, x + (0, 1)), (x, x - (1, 0)), (x, x - (0, 1))).$$

where $B = ((1, 0), (-1, 0), (0, 1), (0, -1))$. Now, each edge receives a *jig* according to a function $\sigma : E \to [q]$. The set of jigs $[q]$ is equipped with an involution $\iota : [q] \to [q]$. We interpret $\iota(j) = j'$ as the jigs j and j' match together, see Figure 2. A *puzzle* is a then function σ such that for all $e \in E^{\text{in}}$,

$$\sigma(e) = \iota(\sigma(\check{e})). \tag{A.1}$$

The case that we have treated previously corresponds to ι equal to the identity.

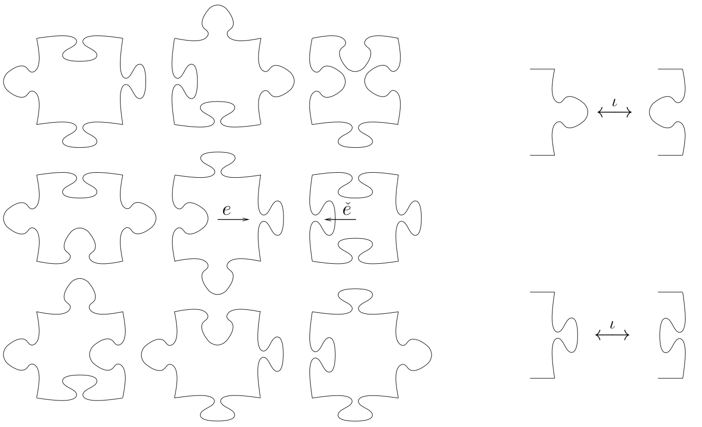

Figure 2: A puzzle with $n = 3$, $q = 4$ and the involution ι.

We now define the way the pieces can be assembled. The *cyclic group* $C_4 \subset S_4$ is the subgroup of permutations generated by $(1\,2\,3\,4)$. Below, if a is a function on E, $s \in S_4$ and $F = (f_1, \cdots, f_4) \in E^4$, we set $a(F) = (a(f_1), \cdots, a(f_4))$ and $F_s = (f_{s(1)}, \cdots, f_{s(4)})$. An *assembly* a is a permutation on E which satisfies :

(i) for every $x \in [n]^2$, there exist a piece $y \in [n]^2$ and $c \in C_4$, such that $a(x + B) = y + B_c$,

(ii) if $y = (1, 1)$, the above permutation c is the identity.

In words, condition (i) says that piece y is assigned a location $x \in [n]^2$ and is rotated by an angle multiple of $\pi/2$. By construction the map which to x assigns y is a bijection of $[n]^2$. Condition (ii) fixes a global orientation to the puzzle. We will say that an assembly is *feasible* if for all $e \in E^{\text{in}}$,

$$\sigma(a(e)) = \iota(\sigma(a(\check{e})).$$

A feasible assembly is a solution of the puzzle : all pieces are in a position where the jigs match. Note that by definition, the identity is a feasible assembly : it gives back the pieces in their original position. We say that a puzzle has *unique vertex assembly* if it has only one feasible assembly (note that without condition (ii), it would only be possible to uniquely assemble the puzzle up to a global rotation by a multiple of $\pi/2$).

Observe that, unlike in a usual jigsaw puzzle, the boundary pieces (pieces in $[n]^2 \backslash [2, n-1]^2$) cannot be distinguished from the other pieces. To recover a usual jigsaw puzzle, we may simply consider the subset of assembly which satisfy the extra condition $a(E^{\text{in}}) = E^{\text{in}}$ (so that edges on the boundary remain on the boundary).

In this new setting, a *random puzzle* is simply obtained by sampling the function σ uniformly on the set of puzzles (functions σ which satisfies (A.1)). Hence, up to the constraint (A.1), the jigs are independent and uniformly distributed. Theorem 1.2 continues to hold on this extended setting. Indeed, it is easy to check that the proof of Theorem 1.2 continues to work if we adapt the definition of the constraint graph (to accommodate the involution).

B Algorithmic aspects - proof of Lemma 5.1

We repeat the statement of the lemma:

Lemma B.1. *If k is as above and $c' = 4/c = 1/50$ in property* (v)*, with high probability, a random puzzle is typical.*

Proof. The first two properties are a consequence of Theorem 2.1. Indeed, from the union bound, Theorem 2.1 implies that with high probability, for any piece $v \in [n]^2$ if $f \in S_k(v)$ then $f(\alpha) = v + \alpha$ for all $\alpha \in \{(0, \pm 1), (\pm 1, 0)\}$. Let us call E, the latter event. By definition, if E holds, any piece has at most one candidate neighborhood and this candidate neighborhood is the neighborhood of the piece in the planted assembly. However, if v is a core piece, $S_k(v)$ is non-empty, hence, if E holds, v has necessary a unique candidate neighborhood. This implies properties (i)-(ii).

We check property (iii). Let $J = \Theta(nk)$ be the number of peripheral edges and let m be the number of peripheral edges which have a non-unique color among the peripheral edges. The probability that two different edges have the same color is $1/q$. Hence, the expectation of m is at most $J(J-1)/q = O((nk)^2/q)$. Since $q \gg n$, from Markov inequality, it implies that with high probability, $m = o(n)$.

We check property (iv). Let us say that pieces v and u have two colors in common, if we can find two jigs of v (say j_1 and j_2), and two jigs of u (say j_3 and j_4) such that $\sigma(j_1) = \sigma(j_3)$ and $\sigma(j_2) = \sigma(j_4)$. The probability that two distinct pieces have two colors in common is at most $6^2/q^2$ if these pieces are not adjacent in the planted puzzle and at most $6^2/q$ if they are adjacent. Hence, the expected number of pairs of peripheral pieces which have two colors in common is at most $O(J^2/q^2 + J/q)$. Since $q \gg n$, the latter is $o(1)$, implying property (iv).

We finally check property (v). It suffices to prove the claim for pieces whose location $(i, j) \in [n]^2$ satisfies that $i + j$ is odd (even) with $c'/2$ instead of c'. We restrict ourselves to those pieces. Note that no two such pieces share any edge. The probability that a specific piece will have two jigs with colors a, b is at most $12q^{-2}$. Therefore, by independence, the probability that there are at least $r \geq 1$ pieces with jigs with colors a, b is at most $n^{2r}(6q^{-2})^r$. We take the union bound over all q^2 pairs of colors. We find that the probability that there is a pair a, b such that there are at least r pieces with jigs with colors a, b is at most

$$6^r n^{2r} q^{-2(r-1)} \leq 6^r n^{2r} n^{-2(r-1)(1+\varepsilon)} = 6^r n^{2-2\varepsilon(r-1)}.$$

For any integer $r > 1 + 1/\varepsilon$, the latter goes to 0 with n. Since $\varepsilon \geq k/c$, we can choose $r = 2 + k/c$. It follows that there at most $(r-1) \leq 2k/c$ pieces with two jigs of a given colors. Since $c = 200$, it implies property (v). □