Supplementary Materials

Peleg Harel
pelegh@post.bgu.ac.il

Ohad Ben-Shahar
ben-shahar@cs.bgu.ac.il

Synopsis

This document constitutes the following supplementary materials

1. The poofs of the noisy mate constraints (for Sec. 4 in the paper)
2. Technical details regarding the synthesis of the datasets used for empirical properties (Sec. 5 in the paper) and experimental evaluation (Sec. 8 in the paper).
3. Proofs for the puzzle properties (for Sec. 5 in the paper).
4. Animated (videos) and extended experimental results (for Sec. 8 in the paper).

Please note that the code for synthesizing and solving crossing cut puzzles, as well as the benchmark dataset, are all available for the scientific community and will be released upon publication of the paper in the authors web site. For the sake of anonymity this code and pointers currently remain concealed but will be included in a form of a URL in the camera-ready version.

1. Noisy puzzles - Proofs and extended discussion

In this section of the Supplementary Materials we extend the corresponding section in the paper and in particular, provide the proofs for the results in its Sec. 4.

Real-world data, its measurement, or its representation, are never completely accurate. Even if the measurement or the digital representation of the pieces were devoid of errors, *real life* crossing cuts puzzles (or geometric puzzles in general) may incorporate deformations to the piece shapes. Such noise can be modelled in many different ways, though one particular appealing is material degradation, and thus piece shrinkage, a process clearly relevant for applications involving physical pieces (e.g. in archaeology).

In order to incorporate material degradation without escaping the crossing cuts framework, we model this process by preserving the number of vertices of each piece, but shifting (i.e., collapsing) each of them *inward* by a random distance that is distributed (in our case, uniformly) in a given range. We note that the particular distribution of such noise may affect certain statistical properties (see Sec. 3 below), but otherwise it is less significant for the reconstruction algorithm proposed below.

Formally, given a noise level ε, a vertex $\overrightarrow{v}_i^j$ of piece p_i is perturbed inwards by a distance $\overrightarrow{\epsilon}_i^j$ that is bounded relative to the puzzle diameter D (distance between furthest vertices). Let ξ be that bound, that sets the absolute noise level at $\varepsilon = \xi \cdot D$. An original piece $p_i = \left\{ \overrightarrow{v}_i^1, \dots \overrightarrow{v}_i^{N_i} \right\}$ ends up as the following *ε-noisy* piece $\tilde{p}_i$

$$\tilde{p}_i = \left\{ \overrightarrow{v}_i^1 + \overrightarrow{\epsilon}_i^1, \dots, \overrightarrow{v}_i^{N_i} + \overrightarrow{\epsilon}_i^{N_i} \right\} \tag{1}$$

$$\text{where} \quad \left\| \overrightarrow{\epsilon}_i^j \right\| \sim \mathrm{U}(0, \varepsilon) \tag{2}$$

$$\text{and} \quad \measuredangle \overrightarrow{\epsilon}_i^j \sim \mathrm{U}\left(\measuredangle \overrightarrow{e}_i^j, \measuredangle \overrightarrow{e}_i^{j+1} \right) \tag{3}$$

where $\measuredangle e_i^j, \measuredangle e_i^{j+1}$ are the angles of the piece edges leaving $\overrightarrow{v}_i^j$ towards the nearby vertices. With this and the distribution of the angle of the perturbation vector $\overrightarrow{\epsilon}_i^j$ is constrained *inward*, i.e., "into" the material rather than along arbitrary direction. Fig. 1A illustrates how such noise could affect the shape of a quadrilateral (4-side) piece.

Naturally, the incorporation of noise affects the validity of our constraints on mating. In particular, the number of potential mates now increases drastically and far from uniqueness, and the implications on a reconstruction algorithm are paramount. In this sense, C_1 and C_2 must be revised, as discussed next.

1.1. $\tilde{C}_1$: Mate length constraint under noise

Since now plausible matings should match edges that have been perturbed differently, the mate length constraint must be relaxed to accommodate these independent perturbations. Let e and e' be the matching edges before applying the noise while $\tilde{e}$ and $\tilde{e}'$ denote

their corresponding ε-noisy edges. It follows that $\tilde{e}$ and $\tilde{e}'$ might have respective lengths $\tilde{L}$ and $\tilde{L}'$ that satisfy $|\tilde{L} - \tilde{L}'| \leq 4\varepsilon$. The maximum error ($4\varepsilon$) can occur when one of the edges is shortened by 2ε and the other is lengthened by 2ε. See Fig. 1B for an illustration of how edges may lengthen even though the deformation represents the *erosion* of material.

1.2. $\tilde{C}_2$: Mate angle constraint under noise

While it is clear that vertices of neighboring pieces may not meet if either sustains noise, and thus may no longer expected to generate two supplementary angles in a strict way, one can still bound the deviation from that ideal behavior. To do so we first analyze the effect of noise on the degree of rotation of any single edge, and then leverage that result for the desired bound on the angles of mating edges under noise.

i. Bound on the rotation of a single ε-noisy edge

Let $e = (\overrightarrow{u}_1, \overrightarrow{u}_2)$ be a piece edge with size $\|\overrightarrow{u}_1 - \overrightarrow{u}_2\| = L$ and coordinates $\overrightarrow{u}_1 = (x_1, y_1)$, $\overrightarrow{u}_2 = (x_2, y_2)$, and assume (without loss of generality) that this edge is aligned with a reference coordinate system such that it lies on the X axis and thus stretches from $\overrightarrow{u}_1 = (0, 0)$ to $\overrightarrow{u}_2 = (L, 0)$. The orientation of this edge is of course $\measuredangle e = 0^o$, as illustrated in Fig. 1A.

Let us now denote by $\tilde{e} = (\overrightarrow{\tilde{u}}_1, \overrightarrow{\tilde{u}}_2) = ((\tilde{x}_1, \tilde{y}_1), (\tilde{x}_2, \tilde{y}_2))$ the same edge after applying the noise. Except for accidental cases (where the vertical translation of the two vertices $\overrightarrow{u}_1$ and $\overrightarrow{u}_2$ due to the noise is identical), the orientation $\measuredangle\tilde{e}$ of edge $\tilde{e}$ will be different than $\measuredangle e$ (as was the case in Fig. 1B, for example). Let $\Delta\Theta_e(L, \varepsilon)$ be the bound on the difference between these two orientations over all possible ε-noisy edges, i.e., over all combinations of $\tilde{e}$ vertices:

$$\Delta\Theta_e(L, \varepsilon) = \max_{\tilde{e}} |\measuredangle\tilde{e} - \measuredangle e| = \max_{\tilde{e}} |\measuredangle\tilde{e}| \qquad (4)$$

To achieve the maximal orientation change $\Delta\Theta_e$ while the vertices of $\tilde{e}$ remain in their respective error zones, it is needed to perturb one of the vertices only horizontally while the other is perturbed vertically as much as possible. This happens when $\tilde{e}$ becomes tangent to the error zone as shown in Fig. 1C and thus the bound is:

$$\Delta\Theta_e(L, \varepsilon) = \begin{cases} \arcsin\left(\frac{\varepsilon}{L-\varepsilon}\right) & L > 2\varepsilon \\ \infty & L \leq 2\varepsilon \end{cases} . \qquad (5)$$

Note that the worst case is assigned to "short" edges ($L \leq 2\varepsilon$) to reflect the possibility they might take arbitrary orientation or simply vanish after the noise. In these cases we set the bound to infinity, representing the fact that the angle constraint cannot contribute useful information.

Eq. 5 requires to know the length of the original ("clean") edge L, but in practice only $\tilde{L}$ can be measured. However, following constraint $\tilde{C}_1$ and Sec. 1.1, it holds that $L \geq \tilde{L} - 2\varepsilon$ and this lower bound can be used as a worst case. We therefore conclude that an ε-noisy edge $\tilde{e}$ with length $\tilde{L}$ might rotate relative to the original "clean" edge no more than

$$\begin{aligned} \Delta\Theta_e(L, \varepsilon) &\leq \Delta\Theta_e(\tilde{L} - 2\varepsilon, \varepsilon) \\ &= \begin{cases} \arcsin\left(\frac{\varepsilon}{\tilde{L}-3\varepsilon}\right) & \tilde{L} > 4\varepsilon \\ \infty & \tilde{L} \leq 4\varepsilon \end{cases} , \end{aligned} \qquad (6)$$

that proves the result presented in the paper.

ii. Bound on the angle difference of two mating edges

Let e and e' be two (clean) mates and denote the corresponding lengths of their edges *before*, *at*, and *after* the mating as L_{-1}, L_0, L_1 and L'_{-1}, L'_0, L'_1, respectively (see Fig. 2A). Let α_1, β_1 and α_2, β_2 be the pairs of supplementary angles these mates form with their adjacent edges at their vertices, as illustrated in Fig. 2A. The mate angle constraint C_2 dictates that

$$\alpha_1 + \beta_1 = \alpha_2 + \beta_2 = \pi \ . \qquad (7)$$

Let $\tilde{\alpha}_i, \tilde{\beta}_i$ $i \in \{1, 2\}$ be the angles corresponding to α_i, β_i after applying the noise (as in Fig. 2B). From Eq. 5 we get

$$|\alpha_1 - \tilde{\alpha}_1| \leq \Delta\Theta_e(L_0, \varepsilon) + \Delta\Theta_e(L_{-1}, \varepsilon) \qquad (8)$$
$$|\alpha_2 - \tilde{\alpha}_2| \leq \Delta\Theta_e(L_0, \varepsilon) + \Delta\Theta_e(L_1, \varepsilon) \qquad (9)$$
$$|\beta_1 - \tilde{\beta}_1| \leq \Delta\Theta_e(L'_0, \varepsilon) + \Delta\Theta_e(L'_{-1}, \varepsilon) \qquad (10)$$
$$|\beta_2 - \tilde{\beta}_2| \leq \Delta\Theta_e(L'_0, \varepsilon) + \Delta\Theta_e(L'_1, \varepsilon) \qquad (11)$$

Combining with the mate angle constraint:

$$\begin{aligned} |\pi - \tilde{\alpha}_1 - \tilde{\beta}_1| \leq\ & \Delta\Theta_e(L_0, \varepsilon) + \Delta\Theta_e(L_{-1}, \varepsilon) \\ & + \Delta\Theta_e(L'_0, \varepsilon) + \Delta\Theta_e(L'_{-1}, \varepsilon) \end{aligned} \qquad (12)$$

$$\begin{aligned} |\pi - \tilde{\alpha}_2 - \tilde{\beta}_2| \leq\ & \Delta\Theta_e(L_0, \varepsilon) + \Delta\Theta_e(L_1, \varepsilon) \\ & + \Delta\Theta_e(L'_0, \varepsilon) + \Delta\Theta_e(L'_1, \varepsilon) \end{aligned} \qquad (13)$$

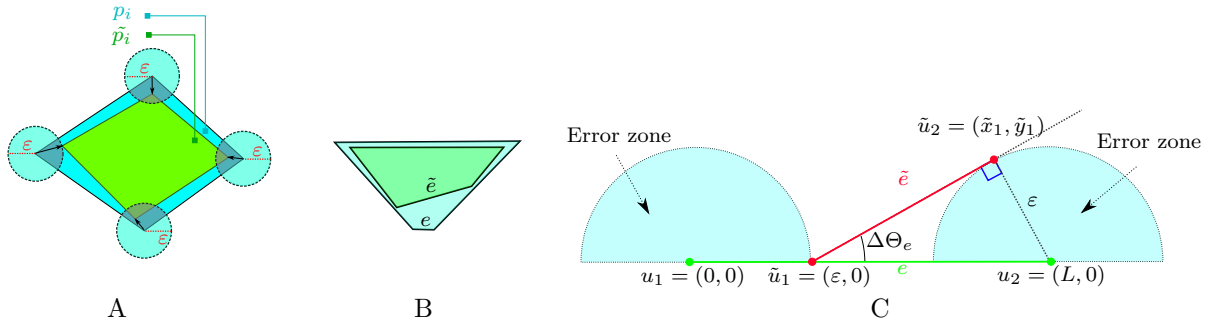

Figure 1: The effect of noise on edge length and orientation. **A:** Each of the vertices of a piece p_i is collapsed inwards along a uniformly distributed direction and as far as a uniformly distributed distance to create the ε-noisy piece $\tilde{p}_i$. **B:** A case where edge e increases in size after the application of noise, even though all the vertices collapsed inwards to end up as $\tilde{e}$. Clearly, $||\tilde{e}||$ is bounded by $||e|| + 2\varepsilon$. **C:** If the clean edge e (in green) stretches (w.l.o.g) from $u_1 = (0,0)$ to $u_2 = (L,0)$, the vertices of the ε-noisy edge bust lie in the corresponding error zones. When considering the angle of the ε-noisy edge $\tilde{e}$ (in red), the worst case occurs when one of the vertices (say, u_1) is only perturbed horizontally by ε, while the other (say, u_2) is perturbed to maximize the rotation, i.e, to a point $\tilde{u}_2 = (\tilde{x}, \tilde{y})$ that makes $\tilde{e}$ tangent to the error zone. This bound is expressed in Eq. 5.

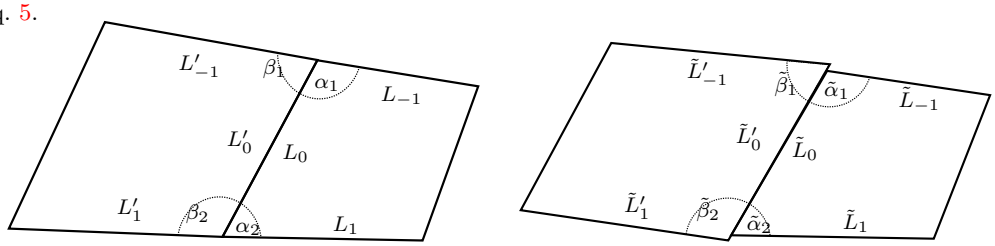

Figure 2: The effect of noise on the mate angle constraint. **A:** Without noise, angles must comply to the original constraint $\alpha_1 + \beta_1 = \alpha_2 + \beta_2 = \pi$ **B:** After applying the noise the ε-noisy angles are affected by the change in orientation in all edges that meet at both vertices of the mating, to result in the bound in Eq. 15.

And finally, we apply the bound in Eq. 6 to reflect the fact that the true edge lengths are unknown. The mate angle constraints under noise thus becomes

$$\begin{aligned} |\pi - \tilde{\alpha}_1 - \tilde{\beta}_1| \leq\ & \Delta\Theta_e(\tilde{L}_0 - 2\varepsilon, \varepsilon) \\ & + \Delta\Theta_e(\tilde{L}_{-1} - 2\varepsilon, \varepsilon) \\ & + \Delta\Theta_e(\tilde{L}'_0 - 2\varepsilon, \varepsilon) \\ & + \Delta\Theta_e(\tilde{L}'_{-1} - 2\varepsilon, \varepsilon) \end{aligned} \tag{14}$$

$$\begin{aligned} |\pi - \tilde{\alpha}_2 - \tilde{\beta}_2| \leq\ & \Delta\Theta_e(\tilde{L}_0 - 2\varepsilon, \varepsilon) \\ & + \Delta\Theta_e(\tilde{L}_1 - 2\varepsilon, \varepsilon) \\ & + \Delta\Theta_e(\tilde{L}'_0 - 2\varepsilon, \varepsilon) \\ & + \Delta\Theta_e(\tilde{L}'_1 - 2\varepsilon, \varepsilon) \ . \end{aligned} \tag{15}$$

As with the mating in the "clean" case, we may refer to the noisy mating constraint as predicates:

$$\begin{aligned} \forall i \in \{1,2\} \quad \tilde{C}_i\left(e_i^j, e_k^l\right) = \text{true} \ \Leftrightarrow \\ e_i^j \text{ and } e_k^l \text{ satisfy constraint } \tilde{C}_i \end{aligned} \tag{16}$$

2. Data synthesis

Since there is no previous work on crossing cuts puzzles, no data or benchmark results exists either. Part of our contribution here is a mechanism for data synthesis, as well as the first public dataset of crossing cut puzzles. Such synthesis tools and dataset facilitate both exploration of statistical properties of such puzzles and the experimental evaluation of reconstruction algorithms.

The synthesis process is based on a computational procedure that receives as input a description of the global shape S (which could be crafted or random; see below) and the crossing cuts that dissect it $Cuts = \{c_1, \ldots c_a\}$. It returns both the *puzzle*, which can be given as input to reconstruction algorithms, and the *ground truth solution* that can be used to evaluate performance of puzzle solvers. As discussed in the paper, the puzzle is a bag of polygonal pieces $P = \{p_1, \ldots p_n\}$, each represented properly by its vertices in some coordinate frame of reference, and the ground truth solution constitutes a representation of the matings M, as well as the Euclidean transformations $((R_1, t_1), \ldots (R_n, t_n))$ that place the pieces correctly in the reconstructed puzzle.

The process of synthesizing crossing cuts puzzles thus constitutes several aspects, all of which are described next for the sake of reproducibility.

2.1. Planar graph representation

Let $S \subseteq R^2$ be polygonal puzzle shape. The first stage of data synthesis is to construct a planar graph $G_{plane} = (\mathcal{V}, \mathcal{E})$ that represents both the boundary of S and the cuts that go through it. Toward that end we first combine both the boundary lines of S (dashed blue lines in Fig.3) and the crossing cuts themselves (dashed red lines in Fig. 3) into one set of lines:

$$\mathcal{C} = Cuts \cup \{\text{edge lines of } S\} \ . \tag{17}$$

The way lines are represented is secondary, but in our case we represent each of them as a triplet (a_1, a_2, a_3), where a_i are the coefficients of the line equation $a_1 x + a_2 y + a_3 = 0$.

The nodes of G_{plane} are the intersection points of any two lines in $\mathcal{C}$ that rest inside or on the border of S (see Fig. 3). Formally, this set of points is defined as follows:

$$\mathcal{V} = \Big\{ i \in S \ \Big| \ \exists (c_1, c_2) \in \mathcal{C} \times \mathcal{C}, \quad (c_1 \neq c_2) \wedge \\ (i \text{ is an intersection of } (c_1, c_2)) \Big\} \ . \tag{18}$$

The set $\mathcal{E}$ of edges of G_{plane} link pairs of nodes that rest on the same line with no other nodes between them:

$$\mathcal{E} = \Big\{ \{i_1, i_2\} \ \Big| \ \exists c \in \mathcal{C}, \\ (i_1, i_2 \in c \cap \mathcal{V}) \wedge ([i_1, i_2] \cap \mathcal{V} = \emptyset) \Big\} \tag{19}$$

where $[i_1, i_2]$ is the line segment between nodes i_1 and i_2.

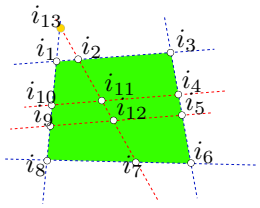

Figure 3: A planar graph extracted from a crossing cuts puzzle of 3 cuts. The green quadrilateral is the global puzzle shape S having the 4 blue boundary lines. It is being cut by 3 red crossing cuts. The nodes of the extracted planar graph are the intersection points inside or on the border of S. Thus, the intersection points $\{i_1 \dots i_{12}\}$ *are* nodes in the graph, but i_{13} is *not*. The edges link two intersection points resting on the same line with no other points in between them. Hence $\{i_3, i_4\}, \{i_3, i_2\}, \{i_4, i_5\}$ *are* edges but $\{i_3, i_5\}$ is *not*.

2.2. Piece extraction and representation

The extraction of the pieces from planar graphs has been addressed in the graph algorithms community and here we employ the optimal algorithm due to Jiang and Bunke [2]. This computational process receives the planar graph G_{plane} from Sec.2.1 and outputs all of the minimal polygonal regions of the graph, each represented by the nodes that delineate it (e.g, $(i_1, i_2, i_{11}, i_{10})$ is one region in the puzzle in Fig. 3).

The main construct in the algorithm is the notion of *wedge*, defined as a pair of edges that meet at a node (e.g., $(\{i_1, i_2\}, \{i_2, i_3\})$ so that no other edge is encountered when rotating the first edge towards the second (e.g. (i_2, i_{11}, i_4) in Fig. 3 is a wedge, but (i_{10}, i_{11}, i_4) is not a wedge). A closed chain of overlapping wedges (e.g $((i_1, i_2, i_{11}), (i_2, i_{11}, i_{10}), (i_{11}, i_{10}, i_1))$ in Fig. 3) defines a region, and thus a puzzle piece (e.g., $(i_1, i_2, i_{11}, i_{10})$ in our example). The sorting scheme that locate the wedge chains was shown to have $O(|\mathcal{E}| \log(|\mathcal{E}|))$ run-time complexity and $O(|\mathcal{E}|)$ memory complexity. Please refer to the original paper [2] for additional details.

We note that once extracted, the pieces are positioned correctly, i.e as they would be positioned in a solved (and 'clean') puzzle. Since the puzzle should include no information about the ground truth position of the pieces, we now center each of them at its center of mass (the average of all vertices) and rotate it by some random angle.

Once the pieces are centered and rotated, we also create a noisy version of each piece by adding to each vertex a random noise vector that obeys the constraints in Eqs. 1 and 3 in Sec. 1 above.

2.3. Extraction of ground truth matings

The original pieces obtained above (before their representation in their own coordinate frame and the application of noise) are positioned in their "correct" place in the solved puzzle. Specifically, any pair of neighboring pieces is positioned such that their mating edges overlap. Hence the extraction of the ground truth matings can be done by finding all identical edges e_i^j and e_k^l that reside in *different* pieces. Formally, if E_m represents the edges of piece p_m (cf. Sec. 2 in the main paper) and thus $E = (\bigcup_{m=1}^{n} E_m)$ is the set of all edges of all pieces, the ground truth matings are

$$M = \Big\{ (e_i^j, e_k^l) \in E \times E \ \Big| \ (p_i \neq p_k) \wedge e_i^k = e_k^l \Big\} \ . \tag{20}$$

2.4. Datasets

We created two datasets using the procedure just described. One was tailored for the empirical exploration

of some statistical properties of crossing cuts puzzles and the other for test and evaluation of crossing cuts solvers including ours, of course)

- **Synthesized puzzles for statistical properties:** Since the statistical properties of crossing cuts puzzles were analyzed for unit circles. The corresponding empirical properties were measured on synthesized puzzles whose shape was a Triacontadigon (i.e., an approximation of a circle as a polygon of 32 sides). The random cuts in this case were select by sampling two angles ϕ_1, ϕ_2 and then passing a line though the corresponding points on the circumference of the circle $(\cos\phi_1, \sin\phi_1)$, $(\cos\phi_2, \sin\phi_2)$.

  Following this procedure we generated a collection of 300 puzzles, 30 puzzles for 10 different numbers of crossing cuts $(10, 20, \ldots, 100)$, as depicted in Fig. 3 of the main paper.

- **Synthesized puzzles for solver evaluation:** Unlike the crafted puzzle shape used for the analysis of properties, the evaluation of our solver (and in the future, others' solvers as well) required randomly shaped (yet convex) puzzles. To achieve this goal we first sampled a random number of random points (between 4 and 50) in some predetermined work space $[0, W] \times [0, H]$ and then computed their convex hull. W, H are given as parameter to the synthesizer and bear very little significance, but in our case we fixed them at $W = H = 100$.

  The random cuts $Cuts = \{c_1, \ldots c_a\}$ were also selected as uniformly distributed random lines in this area, but to ensure they penetrate the random polygon we first selected two random points *inside* polygon and defined the cut as the line that goes between these points.

  While this procedure can be activated on demand, we used it to generate a collection of 175 puzzles, whose global polygonal shape range from 3 to 14 sides, the number of cuts vary from 5 to 35, and number of pieces extends from 14 (in the easier puzzles) to 460 (in the more challenging ones).

  The puzzle dataset used for the solver evaluation is provided here in the Supplementary Materials and further explanations about its format are in the file `Help.txt`. The code for the synthesis and the puzzle solver will be provided upon publication.

3. Puzzle properties

In this section of the Supplementary Materials we extend the corresponding section in the paper and in particular, provide all the proofs for the properties that appear in Table 1 of the paper.

One of the advantages of partially constrained modelled puzzles (cf. Sec. 1 in the main paper) is the better ability to analyze their properties. Since crossing cuts puzzles are results of a stochastic process, their properties are typically probabilistic, but nevertheless can provide insights on both the problem itself and about potential solutions (or limitations thereof). Here we explore such properties both empirically, and when possible, also analytically. In this section we assume that the global puzzle shape is a unit circle (or a polygonal approximation thereof), whose symmetry simplifies some of the analytical analyses.

3.1. Expected cut length

A first measure of interest is the length of a random cut c_i through the global puzzle shape. When the latter is a circle, c_i is determined by two points sampled uniformly on the circumference of the circle. In other words, the cut is determined by the chord between points $\overrightarrow{p_1} = (\cos\phi_1, \sin\phi_1), \overrightarrow{p_2} = (\cos\phi_2, \sin\phi_2)$, where the two angles are uniformly distributed random variables $\phi_1, \phi_2 \sim \mathrm{U}(0, 2\pi)$. The length of cut c_i is therefore another random variable defined by the function $l_i = \|\overrightarrow{p_2} - \overrightarrow{p_1}\|$, and one may seek its expected value.

Since circles are symmetric, without loss of generality we can align the coordinate system parallel to the cut and consider only horizontal chords that lie in the circle's upper half, i.e., when both $\overrightarrow{p_1}$ and $\overrightarrow{p_2}$ have identical positive coordinates, as in Fig. 4A. If we now assume (w.l.o.g) that $\phi_2 > \phi_1$, then $\Theta_i = \phi_2 - \phi_1$ is the central angle of the cut and therefore $l_i = 2\sin(\Theta_i/2)$. Since $\Theta_i \sim \mathrm{U}(0, \pi)$, it follows that the expected length of a unit circle cut is

$$\begin{aligned} E[l_i] &= \int_0^\pi l_i(t) \cdot f_{\Theta_i}(t) dt \\ &= \int_0^\pi 2\sin(\frac{t}{2})\frac{1}{\pi} dt = \frac{4}{\pi} \approx 1.273 \,. \end{aligned} \tag{21}$$

3.2. Probability and total number of cut intersections

Given two uniformly distributed cuts c_1 and c_2, one may seek the probability of their intersection. This question is interesting for understanding how the number of pieces grows with the number of cuts, as intersecting cuts contribute more pieces than non-intersecting ones. Again, for symmetry, we can assume (w.l.o.g) that one of the cuts, say c_1, is horizontal and lying in the upper half of the circle. Let the central angle of c_1 be $\Theta_1 \sim \mathrm{U}(0, \pi)$. c_1 divides the circle to

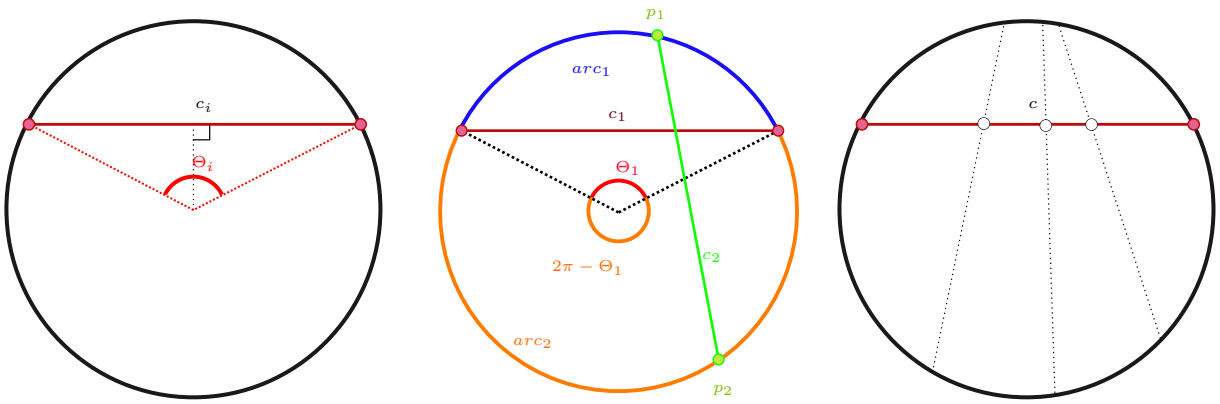

Figure 4: Cut length, intersection probability, and expected number of edges. **A:** A unit circle cut c_i with a central angle Θ_i can be considered w.l.o.g to be horizontal, leading to an expected length as in Eq.21. **B:** Two cuts c_1 and c_2 intersect if and only if the vertices of the second cut (in green) lie in different arcs (in blue and orange) generated by the first cut (in red). The probability of that event is expressed in Eq 23. **C:** The number of edges that emerge from a single cut is one more than the number of intersections one the cut. Here cut c is intersected 3 times to give rise to 4 edges.

two arcs - arc_1 of angle Θ_1 and arc_2 of angle $2\pi - \Theta_1$ (See Fig. 4B).

Denoting the vertices of c_2 as p_1 and p_2, we first note that an intersection between c_1 and c_2 occurs if and only if p_1 belongs to arc_1 and p_2 belongs to arc_2 (or vice versa). Seeking the probability of such an event, let I_{c_1,c_2} be an indicator function for intersection between c_1 and c_2. Clearly, this function depends on the extend (or size) of the two arcs and indeed

$$\begin{aligned} P(I_{c_1,c_2}|\Theta_1) &= 2 \cdot P(p_1 \in arc_1|\Theta_1) \cdot P(p_2 \in arc_2|\Theta_1) \\ &= 2 \cdot \frac{\Theta_1}{2\pi} \cdot \frac{2\pi - \Theta_1}{2\pi} \\ &= \frac{\Theta_1(2\pi - \Theta_1)}{2\pi^2} \end{aligned} \tag{22}$$

It follows that the expected value for the intersection event is

$$\begin{aligned} E[I_{c_1,c_2}] = P(I_{c_1,c_2}) &= \int_0^\pi f_{\Theta_1}(t) \cdot P(I|\Theta_1 = t)dt \\ &= \int_0^\pi \frac{1}{\pi} \frac{t(2\pi - t)}{2\pi^2} dt = \frac{1}{3} . \end{aligned} \tag{23}$$

Hence, somewhat surprisingly, only 1 out of 3 pairs of random unit circle cuts will intersect.

The total number of intersections for a puzzle of a cuts is the sum of all pairs of intersecting cuts, that is

$$N_{intersect} = \frac{1}{2} \sum_{i=1}^{a} \sum_{j \neq i} I_{c_i,c_j}$$

and the expected number of intersections in puzzles with a crossing cuts thus becomes:

$$\begin{aligned} E\left[N_{intersect}\right] &= E\left[\frac{1}{2} \sum_{i=1}^{a} \sum_{j \neq i} I_{c_i,c_j}\right] \\ &= \binom{a}{2} E\left[I_{c_1,c_2}\right] = \frac{a(a-1)}{6} . \end{aligned} \tag{24}$$

3.3. Expected number of edges

Given a crossing cuts puzzle generated by a crossing cuts, we next wish to express the number of piece edges in the entire puzzle. This measure is fundamental to the number of matings and therefore is the substrate of the computational complexity of reconstruction algorithms.

First observe that each edge is a subset of a some cut between two consecutive intersections. In particular, if a cut c_i is intersected k times, the number of edges that emerge from this cut will be $k + 1$ (See Fig. 4a). To obtain the total number of edges N_{edges} in the puzzle one needs to sum up the edges on all cuts, i.e.,

$$\begin{aligned} N_{edges} &= \sum_{i=1}^{a} \left(1 + \sum_{c_j \neq c_i} I_{c_i,c_j}\right) \\ &= a + \sum_{c_i} \sum_{c_j \neq c_i} I_{c_i,c_j} = a + 2N_{intersect} . \end{aligned} \tag{25}$$

Since I_{c_i,c_j} is a random function, so is N_{edges}. We can therefore seek its expected value, i.e., the expected number of edges in the entire puzzle:

$$\begin{aligned} E[N_{edges}] &= E\left[2 \cdot N_{intersect}\right] + a \\ &= 2 \cdot \frac{a(a-1)}{6} + a = \frac{a^2 + 2a}{3} . \end{aligned} \tag{26}$$

3.4. Expected average edge length

With the expected number of edges resolved, we can now seek the expected edge length as the expected ratio between the accumulated edge lengths to their number. Fortunately, the former is simply the summed length of all cuts and thus, if the puzzle constitutes a cuts, we obtain an average edge length of

$$l_{avg} = \frac{\sum_{i=1}^{a} l_i}{N_{edges}} \,. \tag{27}$$

While the expected value of a ratio is *not* the ratio of expected values, it *is* its first order Tylor aproximation [1], which is enough for our purposes. Thus:

$$E[l_{avg}] = E\left[\frac{\sum l_i}{N_{edges}}\right] \approx \frac{E\left[\sum l_i\right]}{E\left[N_{edges}\right]} = \frac{12}{\pi(a+2)} \tag{28}$$

which conforms well with our empirical results shown in Fig5B. The second order Taylor approximation is

$$\begin{aligned} E[l_{avg}] &= E\left[\frac{\sum l_i}{N_{edges}}\right] \\ &\approx \underbrace{\frac{E\left[\sum l_i\right]}{E\left[N_{edges}\right]}}_{\text{First order terms}} \\ &\quad \underbrace{- \frac{Cov(\sum l_i, N_{edges})}{(E[N_{edges}])^2} + \frac{Var(N_{edges}) \cdot E\left[\sum l_i\right]}{(E[N_{edges}])^3}}_{\text{Second order term}} \end{aligned} \tag{29}$$

where the two second order terms are always comparable in size and tend to cancel each other to a diminishing sum, thus facilitating the approximation in Eq. 28 as also exemplified in Fig. 5B.

3.5. Maximum and expected number of pieces

One of the significant properties of a jigsaw puzzles that clearly affects the complexity of their representation (and thus of possible solutions) is its number of pieces. Clearly, even if the number of crossing cuts is set, different cut patterns can create puzzles with varying number of pieces. To estimate this number, and inspired by Moore [3], we use Euler's Formula for planar graphs:

Theorem 1 (Euler's Formula) *If $G = (V, E)$ is any planar graph, then G has $|E| - |V| + 2$ regions where $|E|$ is the number of links in the graph and $|V|$ is the number of nodes.*

Note that in our crossing cuts puzzle case, the number of nodes for Euler's formula is the number of inner intersections ($N_{intersect}$) plus the intersection of the cuts with the puzzle boundary ($2a$), while the number of links is the number of edges (N_{edges}) plus the number of piece sides generated by the cuts in the boundary ($2a$). Using Euler's formula, and applying Eq. 25, we thus get

$$\begin{aligned} N_{pieces} &= \underbrace{(N_{edges} + 2a)}_{|E|} - \underbrace{(N_{intersect} + 2a)}_{|V|} + 2 - 1 \\ &= N_{intersect} + a + 1 \end{aligned} \tag{30}$$

Note that the subtraction of 1 is required since Euler's formula also counts the region outside the puzzle/graph.

With this in mind, we next observe that one extreme case are puzzles where no cut intersect others ($N_{intersect} = 0$) and thus $N_{pieces} = a + 1$ pieces. At the other extreme, every cut intersects all other, yielding $\binom{a}{2}$ intersections and the following quadratic upper bound on the number of pieces

$$\max_{c_i,\ldots c_a} N_{pieces} = \binom{a}{2} + a + 1 = \frac{a^2}{2} + \frac{a}{2} + 1 \tag{31}$$

However, with $N_{intersect}$ being a random variable (that depends on the random cuts), it is more interesting to examine the *expected* number of pieces:

$$\begin{aligned} E\left[N_{pieces}\right] &= E[N_{intersect}] + a + 1 \\ &= \frac{a(a-1)}{6} + a + 1 = \frac{a^2}{6} + \frac{5a}{6} + 1 \,. \end{aligned} \tag{32}$$

This behavior can also be verified empirically, as shown in Fig. 5A. As the number of cuts increases, and when $a \to \infty$, the ratio between the expected and the maximum number of pieces becomes

$$\lim_{a\to\infty} \frac{E[N_{pieces}]}{\max N_{pieces}} = \frac{1}{3} \tag{33}$$

which is the same as the probability for cut intersection found in Sec. 3.2.

3.6. Expected number of edges per piece

Ae discussed in Sec. 2 in the main paper, the crossing cuts puzzle model cuts the puzzle shape into convex polygonal pieces. Clearly, these pieces can have different number of edges and there is no a-priori inherent limit to this number (except the number of cuts, of course).

To explore this property we conducted an empirical evaluation using 30 synthesized crossing cuts puzzles for each of the different number of cuts tested (see the

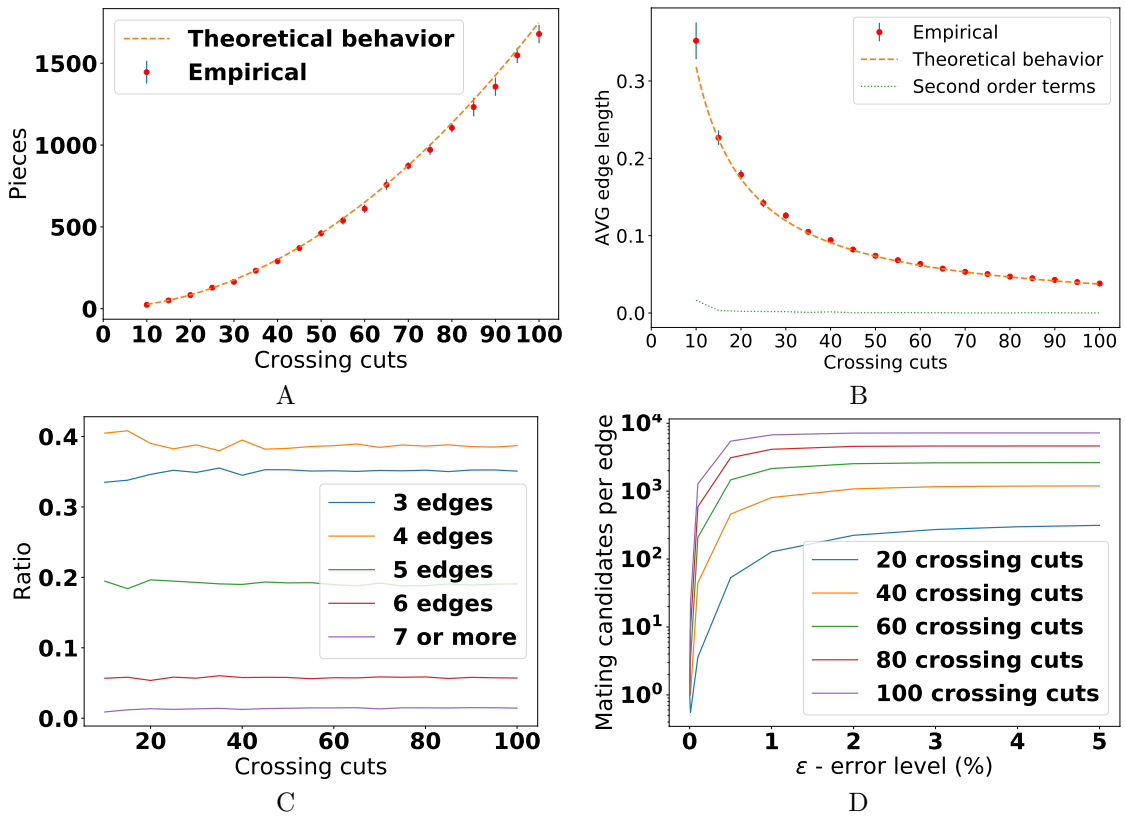

Figure 5: Selected empirical properties of crossing cuts puzzles. **A:** Number of puzzle pieces with growing number of cuts, compared to the theoretical behavior (Eq. 30). Error bars are ±1 SE. **B:** Average edge length with growing number of cuts, compared to the theoretical behavior (Eq. 28). Error bars are ±1 SE. **C:** Expected ratios of pieces with a particular number of edges as a function of the number of crossing cuts. Note how quadrilaterals are always the majority, followed closely by triangular pieces and the less frequent pentagons. These three classes of polygons quickly converge to account for approximately 95% of all pieces.Note how ratios remain invariant to the number of cuts. **D:** The average number of potential mates as a function of noise level. Each graph shows the potential number of mates that satisfy both $\tilde{C}_1$ and $\tilde{C}_2$. The rapid growth indicates the harmful effect of noise, and the numbers converge to twice the number of edges (dashed horizontal lines) since each mating is counted twice, one for each of its participating edges.

Supplementary materials). Empirically, the most frequent pieces are quadrilateral, and the probability to encounter pieces with more than 5 edges is approximately 5% and diminishing quickly. The results are shown in Fig. 5C and demonstrate that the distribution remains relatively stable for increasing number of cuts.

3.7. Number of potential mates per edge

Since any reconstruction algorithm will seek to match the edge of a given piece to edges of other pieces, the complexity of such algorithm will relate intimately to the number of potential mates each edge may have. Clearly, the higher the number of potential mates, the more difficult the identification of the correct one is likely to be. Following the discussion in Sec. 1, this number of potential mates is determined by the two mating constraints $\tilde{C}_1$ and $\tilde{C}_2$ and it is naturally affected by the level of the noise. In fact, a naive extension of the initial algorithm from Sec. 3, that incorporates backtracking when wrong matings are identified, will grow intractably by a factor of $O\left(k^{a^2}\right)$ if the number of potential mates per edge is k.

We explored the expected average number of mates empirically by synthesising puzzle with different numbers of crossing cuts and levels of noise. We counted the average number of possible matings for each puzzle while employing $\tilde{C}_1$ and $\tilde{C}_2$. Not unexpectedly, the results shown in Fig. 5D indicate that the noise level drastically affects the number of potential mates. Observe for example how puzzles with 20 crossing cuts and noise level of 1% generate 100 potential mates per edge.This figure grows two orders of magnitude (close to 10,000) in puzzles with 100 crossing cuts. Here it is worth re-emphasizing that the noise levels in our model are measured relative to the puzzle size, or *diameter*. Thus, considering also the average edge length

(cf. Sec. 3.4), ξ noise level is comparable to $\frac{4 \cdot \xi \cdot \pi(a+2)}{12}$ of average edge length. For a puzzle of 20 crossing cuts (84 pieces on average) and noise level of 1%, the noise is $\approx 10\%$ of the edge length.

Indeed, the high number of potential mates in the presence of noise suggest a similarly high branching factor in a naive "search and backtrack" algorithm, which will clearly become intractable for handling noisy (i.e., realistic) crossing cuts puzzles, even with just modest number of cuts. Our goal is to seek heuristics that make the reconstruction more manageable after all, but at the same time we note that while in this paper we indeed address only *apictorial* crossing cuts puzzles, future research on *pictorial* content, where pictorial constraints can significantly limit the number of potential mates, is critical for successful yet efficient crossing cuts puzzle solvers.

4. Additional experimental results and visualizations

In addition to the quantitative results in the paper, we provide visual demonstrations for the reconstruction process, including the use of the physical spring-mass formulation, the formation of loops, and the merging process, all depicted as animations in video files. The files provided are listed in the in Table 1 below. Note that in all cases the animations are timed for visual convenience and do not represent the actual running times that are listed in the table.

References

[1] Benaroya, H., and Han, S. Probability models in engineering and science. 7

[2] Jiang, X., and Bunke, H. An optimal algorithm for extracting the regions of a plane graph. *Pattern Recognition Letters 14*, 7 (1993), 553–558. 4

[3] Moore, T. L. Using euler's formula to solve plane separation problems. *The College Mathematics Journal 22*, 2 (1991), 125–130. 7

File name	Description	Running time
`positioning.mp4`	A demo of the spring-mass positioning mechanism while it is acting upon hundreds (439) pieces corrupted with significant level of noise (5% of puzzle diameter, and more than 50% of average edge length).	Positioning: 1.26s
`reconstruction0.mp4`	A demo of the full reconstruction process for a puzzle of 10 cuts, 40 pieces, $\xi = 0.1\%$, and $\bar{\xi} = 1\%$.	Locate loops : 1e-3s Score loops : 8e-4s Merge loops : 3e-5s Positioning: 9e-5s
`reconstruction1.mp4`	A demo of the full reconstruction process for a puzzle of 8 cuts, 21 pieces, $\xi = 0.1\%$, and $\bar{\xi} = 1\%$.	Locate loops : 1e-3s Score loops : 5e-3s Merge loops : 8e-5s Positioning: 9e-5s
`reconstruction2.mp4`	A demo of the full reconstruction process for a puzzle with a significant level of noise. The puzzle is of 6 cuts, 19 pieces, $\xi = 1\%$, and $\bar{\xi} = 8\%$.	Locate loops : 2e-3s Score loops : 2.6s Merge loops : 1e-3s Positioning: 1e-4s
`reconstruction3.mp4`	A demo of the full reconstruction process for a larger puzzle of of 35 cuts, 400 pieces, $\xi = 0.01\%$, and $\bar{\xi} = 0.6\%$.	Locate loops : 1.87s Score loops : 39s Merge loops : 1e-2s Positioning: 1e-3s

Table 1: Description of video files.