Jigsaw Puzzles, Edge Matching, and Polyomino Packing: Connections and Complexity*

Erik D. Demaine, Martin L. Demaine

MIT Computer Science and Artificial Intelligence Laboratory,
32 Vassar St., Cambridge, MA 02139, USA, {edemaine,mdemaine}@mit.edu

Dedicated to Jin Akiyama in honor of his 60th birthday.

Abstract. We show that jigsaw puzzles, edge-matching puzzles, and polyomino packing puzzles are all NP-complete. Furthermore, we show direct equivalences between these three types of puzzles: any puzzle of one type can be converted into an equivalent puzzle of any other type.

1. Introduction

Jigsaw puzzles [37,38] are perhaps the most popular form of puzzle. The original jigsaw puzzles of the 1760s were cut from wood sheets using a hand woodworking tool called a jig saw, which is where the puzzles get their name. The images on the puzzles were European maps, and the jigsaw puzzles were used as educational toys, an idea still used in some schools today. Handmade wooden jigsaw puzzles for adults took off around 1900. Today, jigsaw puzzles are usually cut from cardboard with a die, a technology that became practical in the 1930s. Nonetheless, true addicts can still find craftsmen who hand-make wooden jigsaw puzzles. Most jigsaw puzzles have a guiding image and each side of a piece has only one "mate", although a few harder variations have blank pieces and/or pieces with ambiguous mates.

Edge-matching puzzles [21] are another popular puzzle with a similar spirit to jigsaw puzzles, first appearing in the 1890s. In an edge-matching puzzle, the goal is to arrange a given collection of several identically shaped but differently patterned tiles (typically squares) so that the patterns match up along the edges of adjacent tiles. Typical patterns range from salamanders to frogs to insects, but in their simplest form each edge of a tile is simply colored one of several colors, and adjacent tiles must be colored identically along their common edge. In a harder abstract form of edge-matching puzzles, edges also have a sign (+ or −) and adjacent tiles must have opposite sign (like magnetism). This constraint represents two different parts to a pattern that must come together—preventing, for example, two heads or two tails from matching. Fig. 1 shows a concrete example of such a puzzle that we designed in honor of Jin Akiyama. Edge-matching puzzles are challenging compared to standard jigsaw puzzles because there is no global image to guide the puzzler, nor do two pieces "fitting together" guarantee that they should

* A preliminary version of this paper was presented at the *Gathering for Gardner 6*, Atlanta, March 2004.

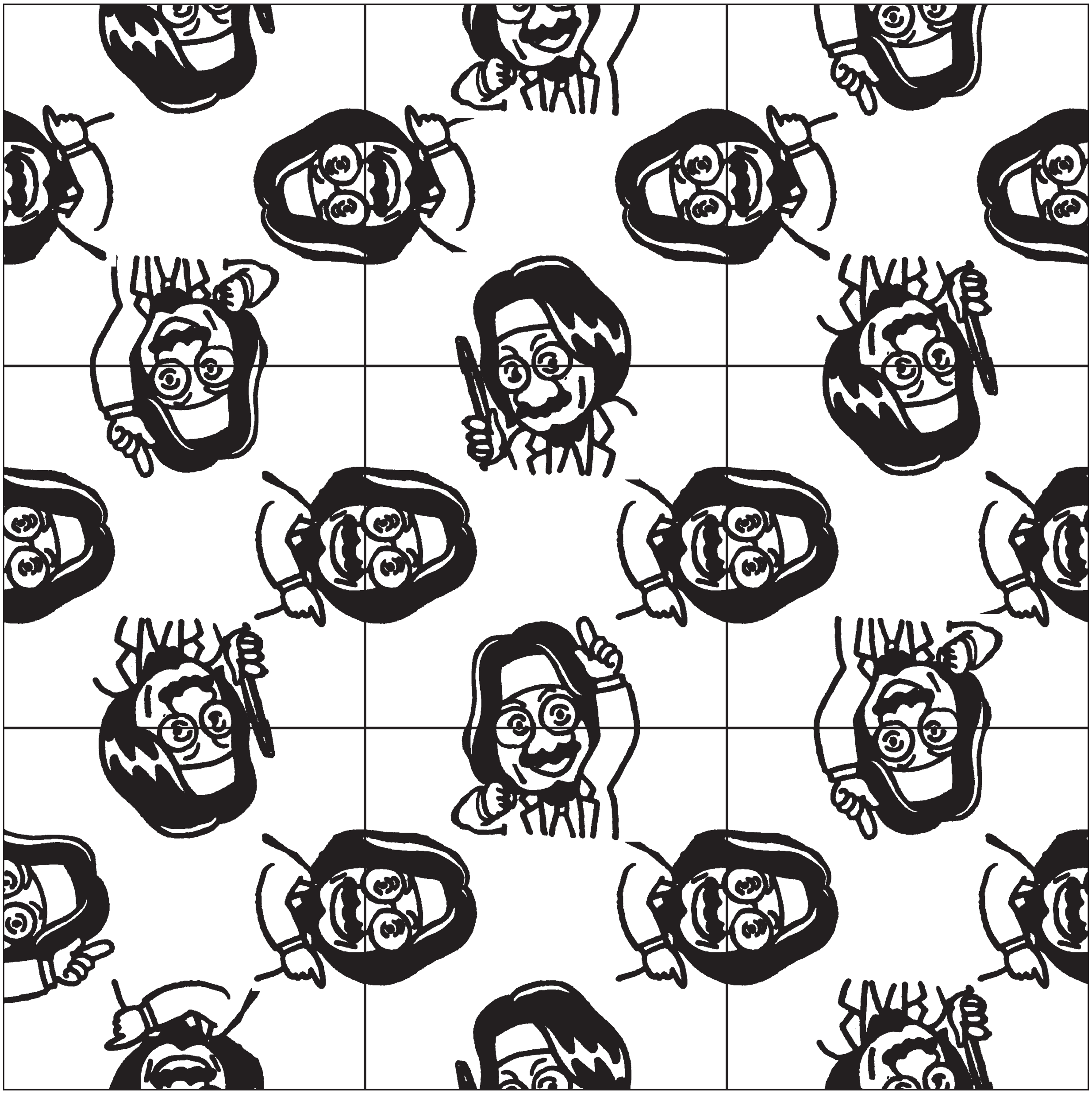

Fig. 1. A new (signed) edge-matching puzzle in honor of Jin Akiyama. The cartoon renditions are from [1].

be together. Only completing the entire solution guarantees correctness of any local part of the solution.

Polyform packing puzzles also have a jigsaw-like flavor. These puzzles were popularized by the recent Eternity puzzle [31, 29], whose solution won two mathematical puzzlers £1,000,000. Polyform packing puzzles were introduced by Golomb around 1965 [20] when he defined the first type of polyform, polyominoes. A polyomino arises from gluing unit squares together edge-to-edge. In general, a polyform arises from edge-to-edge gluing of several copies of a simple shape, such as a square, an equilateral triangle, or an equilateral triangle cut in half (as in Eternity). In a polyform packing puzzle, the goal is to pack a given collection of several polyforms into precisely a given shape—the target shape—such as a larger rectangle, rhombus, or dodecagon (as in Eternity). Polyform packing puzzles are in some ways an ultimate form of jigsaw puzzle: not only is there no guiding image and two pieces fitting together says nothing about whether they are together in the final solution, but also two pieces can fit together in several different ways.

In this paper we study how efficiently a computer can solve all three of these types of puzzles. We show that, computationally, these three types of puzzles are all effectively the same: a puzzle of one type can be converted into an equivalent puzzle of each of the other types, with a small blowup in puzzle size. The *equivalence* between the two puzzles is strong: every solution in one puzzle corresponds to a solution in the other puzzle, by a simple, efficient, and invertible conversion. We prove the equivalence by a circular series of reductions illustrated in Fig. 2. Furthermore, we show that these types of puzzles are computationally intractable—NP-complete—implying that, likely, no efficient algorithm can solve these types of puzzles in general. These results confirm the challenge that humans have had in solving these puzzles, and justify the exhaustive search that has seemed necessary when solving these puzzles via computer. (For example, see [31] for how the winners solved the Eternity puzzle.) Our proofs are fairly simple but appear not to have been observed before.

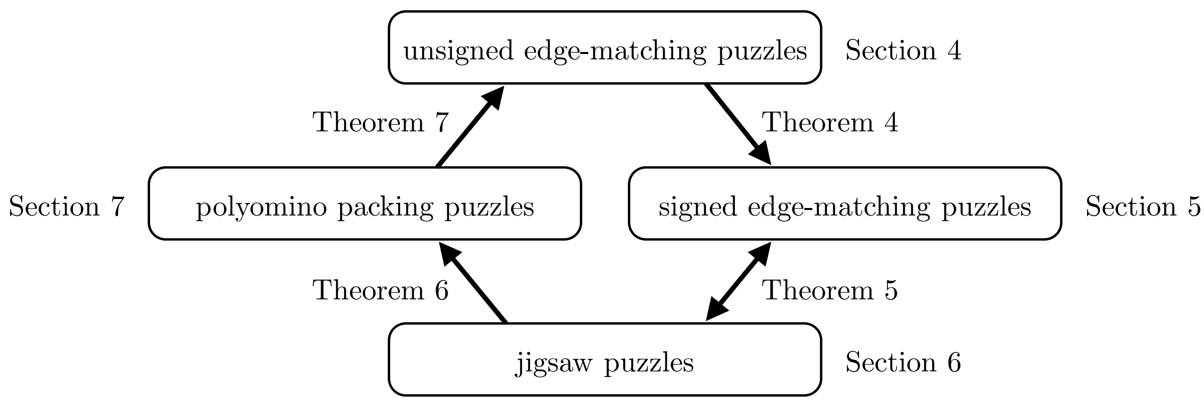

Fig. 2. Our series of reductions between puzzle types.

Related work. There are several results related to ours. Perhaps the best-known result, proved by Berger [5] and described by Martin Gardner [15], is that edge-matching puzzles are undecidable—no algorithm can solve them in general—when given an infinite number of copies of each tile, and the goal is to fill the entire plane. Our puzzles have only finitely many copies of each tile, which forces the problem to be decidable simply by trying all possible tile placements. In the context of a finite (polynomial-size) rectangular target shape, Berger's result implies that the problem is NP-complete when given arbitrarily many copies of each tile, or equivalently, if not all of the given tiles have to be used in the puzzle. This result is also in unpublished 1977 work of Garey, Johnson, and Papadimitriou [18, p. 257], and used in Levin's theory of average-case completeness [25]. In contrast, the edge-matching puzzles we consider must use exactly the tiles given.

Polyomino packing puzzles are NP-complete when the target shape is complicated (a polyomino with holes) and the pieces are all identical (either 2×2 squares, 1×3 rectangles, or 2×2 L shapes) [28]. In contrast, the polyomino packing puzzles we consider have a simple target shape (a larger square) and the problem is all about using the different tiles given. Polyomino packing puzzles are also known to be weakly NP-complete when the target shape is simple (a rectangle) and the pieces are $1 \times k$ rectangles for various exponentially large values of k [22]. In unpublished work, Baxter [4] proves that polyomino packing puzzles are (strongly) NP-complete when the pieces are polyominoes of polynomial area and the target shape is a rectangle.[1] In contrast, the pieces in our polyomino packing puzzles are small squares with small bumps; in particular, all pieces have polylogarithmic area, an exponential improvement.

The shape-matching community has studied computational solutions to jigsaw puzzles extensively [12, 19, 23, 30, 40]. This work generally assumes that, if two shapes locally fit together well, they will be together in the final global solution. In contrast, we consider a harder form of jigsaw puzzles where some pieces have ambiguous mates. This ambiguity is necessary to make a computationally interesting puzzle: if it is locally possible to tell whether two pieces fit together in the final solution, then the naïve solution of trying to join together all pairs of pieces solves a puzzle in polynomial time.

Most packing puzzles demand *exact* packings, which use every given piece to fill exactly the target shape, with no overlap or blank space.[2] Of course, for this goal to be possible, the total area of the given pieces must equal the area of the target shape. When the packing may leave blank space in the target shape, orthogonally packing rectangles or squares of various polynomial sizes into a given square is NP-complete [26, 24]. (In fact, we use this result in Section 3.) These results assume that the rectangles pack orthogonally, but Erdős and Graham [11] have shown that this can be suboptimal, even with squares all of the same size; see also [13]. If we additionally allow the target shape to consist of multiple squares of specified size, and the goal is to minimize the number of such squares, we obtain a 2D generalization of the classic bin-packing problem. Epstein and Stee [10] have developed constant-factor approximation algorithms and inapproximability results for this problem when the shapes to be packed are squares of various sizes.

Back to exact packings, several researchers have considered various forms of exact packings of integer squares into a larger integer square. One of the earliest questions along these lines is whether it is possible to pack the k smallest integer squares—1×1, 2×2, …, $k \times k$—exactly into a larger square. The area constraint is $1^2 + 2^2 + \cdots + k^2 = n^2$, which has a unique integer solution of $k = 24$ and $n = 70$ [36]. However, there is no matching square packing [6]; see [39]. Another problem, called *Mrs. Perkins's quilt* by Conway [8, 33, 16], asks for the fewest squares whose side lengths have no overall common divisor larger than 1 that exactly pack a given $n \times n$ square, as a function of n. A final problem, *squaring the square* [7, 34, 35, 9, 14], asks for distinct squares that exactly pack a larger square. This problem has also been considered in the context of triangles [32].

Akiyama et al. [2, 3] considered the related dissection problem of dividing a square into pieces that can be re-assembled into two, three, …, or k squares. They show that such a dissection is possible with as few as $2k + O(1)$ pieces, and that this is the best bound possible among the family of "purely recursive dissections".

Simple upper bounds on complexity. All of the problems we consider are in the complexity class NP, because it is easy to verify the validity of a packing or tiling in polynomial time. Thus, to prove NP-completeness of a problem, it suffices to prove *NP-hardness*, i.e., to prove that the problem is at least as hard as all problems in NP.

[1] In fact, it was Baxter's message that inspired us to write this paper.

[2] In computer science, exact packings are often called *tilings*. We avoid this terminology to prevent confusion with terms such as "polyomino tiling" which usually refers to tiling the entire plane.

Another basic result is that the puzzles we consider need sufficiently many different piece types to be hard. If a jigsaw puzzle, edge-matching puzzle, or polyform packing puzzle has only a constant number c of different piece types, then the problem becomes *sparse*: the number of different puzzles with n pieces is bounded by a polynomial n^c. Mahaney [27] proved that, if a sparse problem is NP-complete, then P = NP, which is unlikely. (Of course, we could consider compressing such sparse puzzles to encode in binary the number of copies of each piece—but such compressed problems seem difficult to study, or at least of a different nature.)

Most likely, we need polynomially many piece types before the puzzles can be NP-complete. Our jigsaw and edge-matching puzzles are essentially tight against this bound, up to the constant degree of the polynomial. For polyomino packing puzzles, this assumption implies that the area of some of the pieces must be $\Omega(\log n)$, while our construction uses pieces of area $O(\log^2 n)$, leaving an interesting gap for future study.

2. Rectangle Packing Puzzles

We start by considering the computational complexity of a simple form of polyomino packing puzzle, in which every piece is a $1 \times x$ rectangle for some integer x. Kempe [22] proved that such puzzles are *weakly* NP-complete: for his hardness result to apply, the sizes of the rectangles (the values of x) must be exponential in the number of pieces. We prove that rectangle packing puzzles are *strongly* NP-complete to solve, even when rectangle sizes are polynomial in the number of pieces:

Theorem 1. *It is (strongly) NP-complete to decide whether n given rectangular pieces—sized $1 \times x_1$, $1 \times x_2$, $\ldots$, $1 \times x_n$ where the x_i's are positive integers bounded above by a polynomial in n—can be exactly packed into a specified rectangular box of area $x_1 + x_2 + \cdots x_n$.*

Although the proof of this theorem is simple, and only slightly harder than Kempe's proof [22], it provides the core idea followed in all of our (more complicated) hardness proofs.

The NP-hardness proof is a reduction from 3-partition. In other words, we show that solving rectangle packing puzzles is at least as hard as the (NP-complete) problem 3-partition by showing how to convert any instance of the 3-partition problem into an equivalent rectangle packing puzzle.

In the 3-partition problem [17], [18, pp. 96–105, 224], we are given a set of $3m$ positive integers $A = \{a_1, a_2, \ldots, a_{3m}\}$, where the a_i's are bounded above by a polynomial in m, and the goal is to partition the set A into m triples such that every triple has the same sum. Specifically, each triple must sum to $\Sigma = \frac{1}{m}(a_1 + a_2 + \cdots + a_{3m})$. Furthermore, we may assume that each a_i is strictly between $\Sigma/4$ and $\Sigma/2$, so that any set of a_i's summing to Σ must have size exactly 3. Some sets A have a *3-partition* (a partition into triples of equal sum), while other sets A have no 3-partition; it is distinguishing between these two cases that is NP-hard.

We show how to efficiently convert a set A into a rectangle packing puzzle such that the puzzle is solvable precisely if the set A has a 3-partition. Refer to Fig. 3. The bulk of the conversion is a simple addition: for each integer a_i in A, there is one $1 \times (a_i + m)$ rectangular piece. The rectangular box has size $m \times (\Sigma + 3m)$.

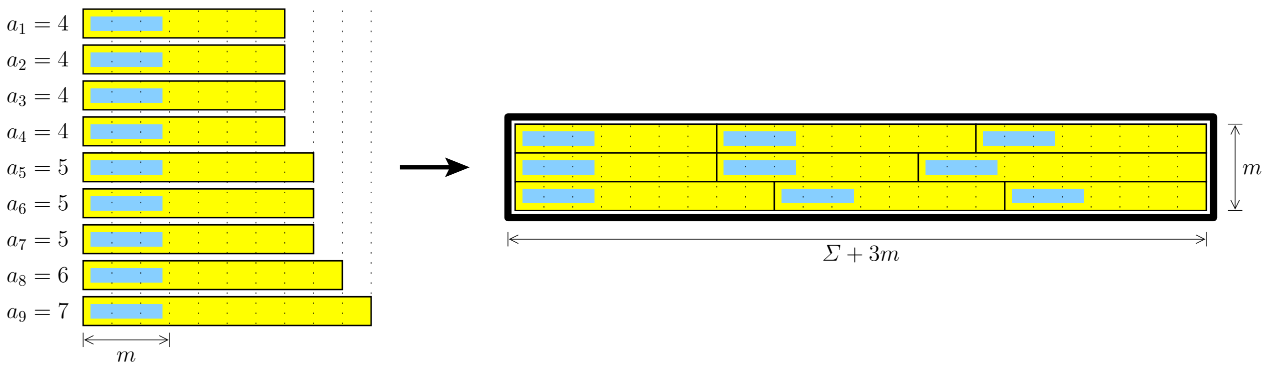

Fig. 3. Rectangle packing puzzle representing 3-partition.

Any 3-partition of A determines one way to pack the rectangles: put the three rectangular pieces corresponding to a triple of integers into one row of the rectangular box. Because each triple of a_i's sums to Σ, the three rectangular pieces in each row fill exactly the $\Sigma + 3m$ of horizontal space in the box. There are exactly m triples, so this assignment fills all m rows of the box.

Why is this the only way to pack the rectangles? More precisely, why does every packing determine a 3-partition of A? First, because the area of the rectangular pieces equals the area of the rectangular box, the pieces must be placed on the integer grid of the box, and in particular can be rotated only by multiples of 90°. (This fact follows by induction: every convex corner of what remains of the box must be filled by a corner of a piece.) Second, because each piece is at least $m+1$ units long, yet the box is only m units tall, every piece must be placed horizontally (unrotated). Third, because each a_i is strictly between $\frac{1}{4}\Sigma$ and $\frac{1}{2}\Sigma$, each piece length $a_i + m$ is in particular strictly between $\frac{1}{4}(\Sigma + 3m)$ and $\frac{1}{2}(\Sigma + 3m)$, so any filled row of the box must consist of exactly three pieces. Therefore, any packing of all rectangular pieces into the box determines a 3-partition of A, concluding the proof of NP-hardness of rectangle packing puzzles.

3. Square Packing Puzzles

Among the simplest of polyomino shapes are $1 \times x$ rectangles and $x \times x$ rectangles (squares). To complement the result of the previous section, we next prove NP-hardness for square packing puzzles:

Theorem 2. *It is (strongly) NP-complete to decide whether n given square pieces—sized $x_1 \times x_1$, $x_2 \times x_2$, …, $x_n \times x_n$ where the x_i's are positive integers bounded above by a polynomial in n—can be exactly packed into the square box of area $x_1^2 + x_2^2 + \cdots x_n^2$.*

The proof of this theorem does not follow the same structural outline as any of the other proofs; in particular, our reduction is not from 3-partition. Thus, this result is a bit of a diversion from the main line. It is, however, a short diversion, because the proof follows almost immediately from a known result.

Motivated by applications in job scheduling on partitionable mesh supercomputers, Leung et al. [24] proved that it is strongly NP-complete to orthogonally pack given squares into a given square. (Previously, Li and Cheng [26] proved the same result when the target shape is allowed to be any given rectangle.) The main difference between their problem and ours is that their packings are not exact: the area of the target shape can be larger than the total area of the pieces. Fortunately, they also force the packing to be orthogonal—pieces cannot be rotated—and all of their packings lie on the integer grid of the target square. These properties make it possible to reduce from their problem to ours, proving that our problem is at least as hard and thus NP-hard.

Suppose we are given an instance of integral orthogonal square (nonexact) packing of squares sized $a_1 \times a_1$, $a_2 \times a_2$, $\ldots$, $a_n \times a_n$ into a $b \times b$ square. We construct an instance of an (exact) square packing puzzle simply by adding $b^2 - a_1^2 - a_2^2 - \cdots - a_n^2$ copies of the 1×1 square to the existing collection of m pieces. The total area of the resulting piece set equals the area of the square box. As argued in Section 2, this property forces pieces to remain orthogonal and on the integer grid of the box. Furthermore, the additional squares do not prevent any previous integral orthogonal packings of the other pieces, because they can be placed in the gaps of any such packing. Therefore, the new piece set has an integral orthogonal packing precisely if the original piece set has an exact packing, proving NP-hardness of square packing puzzles.

4. Unsigned Edge-Matching Puzzles

Neither result from the previous two sections is particularly satisfying as establishing hardness of polyomino packing puzzles, because both results require pieces with area polynomial in the number of pieces. In Section 7, we will show that polyomino packing remains NP-complete even with pieces of polylogarithmic area. We obtain this result by simulating edge-matching puzzles, so we first turn to the hardness of such puzzles. Specifically, in this section we prove that it is NP-complete to solve an unsigned edge-matching puzzle:

Theorem 3. *It is NP-complete to decide whether n given unit-square tiles, each with a specified color on each edge, can be placed into a $\sqrt{n} \times \sqrt{n}$ square box such that all adjacent tiles have matching colors along their common edge. The problem remains NP-complete if the puzzle has $4\sqrt{n}$ colors each occurring exactly once (which must thus be against the box boundary in any solution).*

The proof of NP-hardness essentially mimics the reduction from 3-partition to rectangle packing puzzles from Section 2. As in that proof, we show that solving unsigned edge-matching puzzles are at least as hard as 3-partition by showing how to convert any instance of the 3-partition problem into an equivalent unsigned edge-matching puzzle.

The puzzle has one main structural element, the *frame*, that divides the $(2+\max\{m, \Sigma\})$ $\times$ $(2 + \max\{m, \Sigma\})$ square box into a rectangular hole of size $m \times \Sigma$. See Fig. 4. Each exterior edge of this frame uses a unique color, which has no matching tile, forcing these tiles to be on the exterior of the square box. The shared edge between every two adjacent tiles also uses a unique color, forcing these two tiles to be adjacent, because this color appears only twice and because all exterior edges have been accounted for. These constraints rigidly determine the frame; all that remains is to fill the inside of the frame. The horizontal interior edges of the frame are all a common color, %. The vertical interior edges of the frame are all a different common color, $.

The puzzle has a_i tiles to represent each number a_i in S. See Fig. 5. All of these tiles are colored % on their top and bottom edges. One of the tiles is colored $ on its left edge, and another tile is colored $ on its right edge. All other left and right edges are colored a single color that is unique to this a_i. Thus, all but two of the tiles are identical. Because

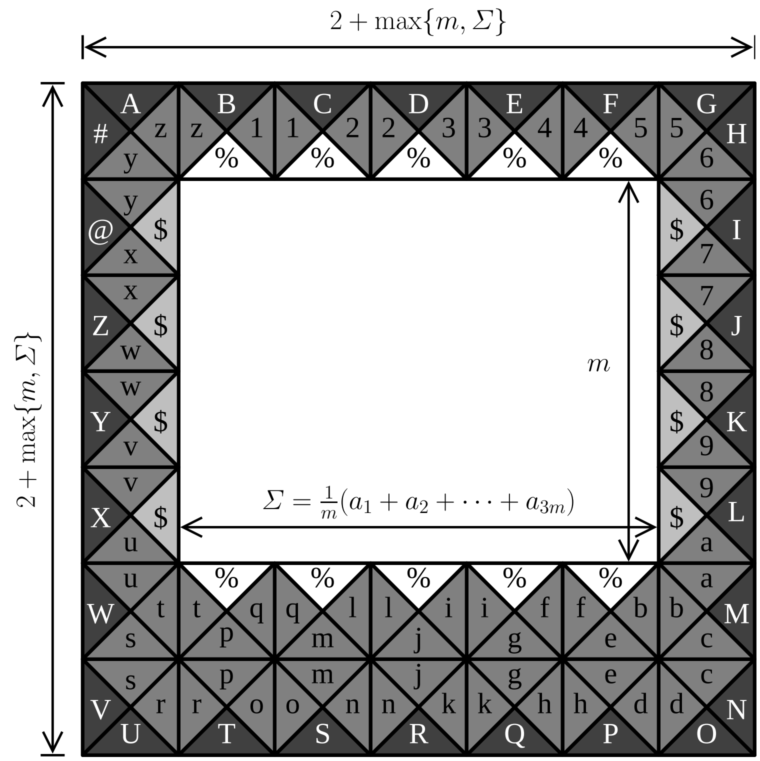

Fig. 4. Example of the frame construction. Colors are denoted by symbols; shading just distinguishes different classes of colors.

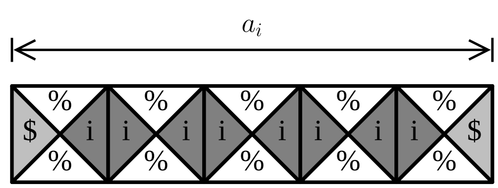

Fig. 5. Example of representing a number a_i by a $1 \times a_i$ rectangle of tiles.

of the color unique to a_i, the only way that these tiles can fit together is into a $1 \times a_i$ rectangle, colored % on the long sides and $ on the short sides. The coloring of the interior frame edges forces these rectangles to be horizontal.

Therefore, the unsigned edge-matching puzzle becomes a problem of packing horizontal $1 \times a_i$ rectangles into an $n \times \Sigma$ rectangle. As argued in Section 2, this problem is equivalent to the original 3-partition problem, concluding the proof of NP-hardness of unsigned edge-matching puzzles.

This NP-hardness reduction has some differences with the reduction for rectangle packing puzzles from Section 2. The main difference is the added frame, which allows us to force the box to be a square, a desired property for elegance of the theorem statement. The frame also provided an alternative way to force the rectangles to remain horizontal, so we did not have to add m to the length of each rectangle. Although we could have done without the outermost layer of the frame in this hardness proof, opting for just a uniquely constructable $(\max\{m, \Sigma\} - \min\{m, \Sigma\}) \times \max\{m, \Sigma\}$ rectangle, we will find the unique colors on the boundary of the construction useful for jigsaw puzzles and polyomino packing in Sections 6 and 7.

5. Signed Edge-Matching Puzzles

Many edge-matching puzzles fall into a different category from the problem analyzed in the previous section. Specifically, in *signed* edge-matching puzzles, tile edges have a *sign* of + or − in addition to a color. Two adjacent such tiles match if their common edges have the same color and opposite sign. This behavior matches magnetism, for example. This behavior is also quite natural in many real puzzles because they represent the color by an image of an insect, amphibian, etc. cut across two tiles, and unless the image happens to be cut along a line of symmetry, the two sides are different (e.g., the head and the tail) and thus must come together in opposite pairs to form a complete image as the puzzle requires.

We show that signed edge-matching puzzles can represent unsigned puzzles:

Theorem 4. *Every unsigned edge-matching puzzle with c colors and t tile types has an equivalent signed edge-matching puzzle with $2c + t$ colors and $4t$ tile types.*

Fig. 6 illustrates the conversion. We replace each tile in the unsigned edge-matching puzzle with four signed tiles that uniquely combine into a 2×2 square or *supertile*. Four colors unique to this unsigned tile type force the unique combination into this signed supertile. Each edge of the supertile has a double label consisting of the same color, negative and then positive in clockwise order around the supertile. Hence, any two compatible unsigned tiles produce two compatible signed supertiles, so any solution to the unsigned puzzle corresponds to a solution to the signed puzzle. Conversely, because any solution to the signed puzzle must form the supertiles, the solution must align supertiles on the even integer grid of the box, leading to a solution to the unsigned puzzle. Therefore, the unsigned and signed edge-matching puzzles are equivalent.

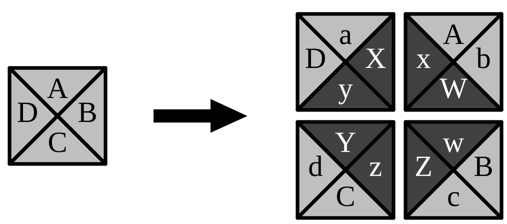

Fig. 6. Representing an unsigned tile by four signed tiles. Letters denote colors, with case denoting sign. The colors w–z are unique to this tile type.

This conversion also serves as a reduction, so an immediate consequence by Theorem 3 is NP-hardness of signed edge-matching puzzles:

Corollary 1. *It is NP-complete to decide whether n given unit-square tiles, each with a specified color and sign on each edge, can be placed into a $\sqrt{n} \times \sqrt{n}$ square box such that all adjacent tiles have matching colors and opposite signs along their common edge.*

This result can also be proved directly using a variation of the reduction for unsigned edge-matching puzzles from Section 4, avoiding the constant-factor losses in the number of colors, tile types, and box size. Namely, we assign positive signs to north and east edges of tiles, and we assign negative signs to south and west edges of tiles. Here we exploit that tiles never need to rotate relative to our figures in order to solve these edge-matching puzzles.

6. Jigsaw Puzzles

There are many different kinds of jigsaw puzzles. In this section, we consider one of the simplest kinds: each piece is a square augmented on each edge with a tab, with a pocket, or not at all. Tabs and pockets can have arbitrary shapes (of sufficiently small size to not cause overlap), and only matching tabs and pockets fit together. Unaugmented edges can fit against each other or against the boundary of a rectangular box.

Theorem 5. *Every signed edge-matching puzzle with an $x \times y$ box, xy tiles, t tile types, c colors, and $2(x+y)$ colors each occurring exactly once (which must thus be against the box boundary in any solution) has an equivalent jigsaw puzzle with the same $x \times y$ box, xy pieces, at most t piece types, and $c - 2(x+y)$ pocket shapes.*

Conversely, every jigsaw puzzle with an $x \times y$ box, xy pieces, t piece types, and c pocket shapes has an equivalent signed edge-matching puzzle with the same $x \times y$ box, xy tiles, at most $t + 2(x+y)$ tile types, $c + 2(x+y)$ colors, and $2(x+y)$ uniquely occurring colors.

The heart of the equivalence is straightforward: color corresponds to tab/pocket shape, positive sign corresponds to a tab, and negative sign corresponds to a pocket. See Fig. 7 (left and middle). The only difficulty is properly dealing with the boundary. The signed edge-matching puzzles considered by the theorem, use $2(x+y)$ uniquely occurring colors to force which edges form the box boundary. Jigsaw puzzles use $2(x+y)$ flat edges to force which edges form the box boundary. Therefore we simply define a correspondence between these features, and we have the desired equivalence.

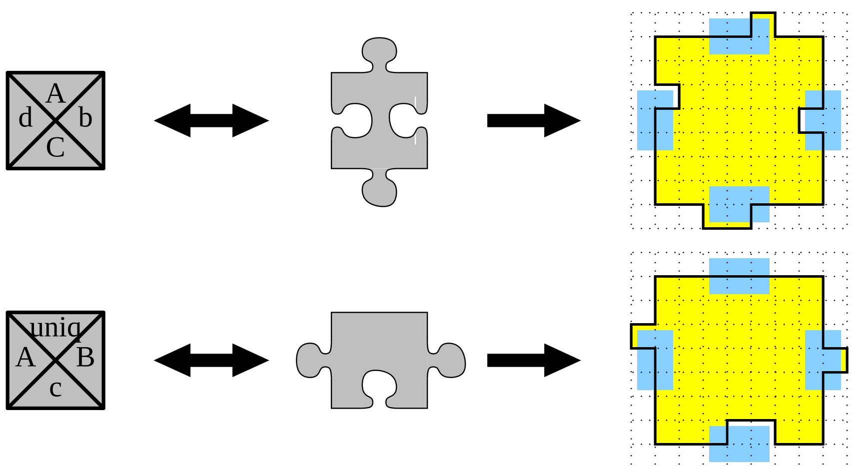

Fig. 7. Representing a signed edge-matching tile by a jigsaw piece or by a polyomino piece. Letters denote colors, with case denoting sign.

An immediate consequence of this equivalence and Corollary 1 is NP-hardness of jigsaw puzzles:

Corollary 2. *It is NP-complete to decide whether n given jigsaw pieces fit together to form a $\sqrt{n} \times \sqrt{n}$ square box.*

7. Polyomino Packing Puzzles

The penultimate in our series of reductions converts jigsaw puzzles of the type considered in the previous section into equivalent polyomino packing puzzles. This result is interesting because it establishes a close relationship between the two types of puzzles, but also because it proves a stronger form of NP-completeness of polyomino packing puzzles. We already saw in Sections 2 and 3 that such puzzles are NP-complete, but these results required pieces with area polynomial in the number of pieces. This section establishes NP-completeness even when the pieces have polylogarithmic area.

Theorem 6. *Every jigsaw puzzle with an $x \times y$ box, xy pieces, t piece types, and c pocket shapes has an equivalent polyomino packing puzzle with an $xs \times ys$ box, xy pieces, t piece types, and every piece fits within an $(s+1) \times (s+1)$ rectangle, where $s = 4 + \lceil \lg(c+1) \rceil$.*

The conversion simulates the shape of a tab or pocket by a binary string of unit-square tabs or pockets along the middle of an edge of an $s \times s$ square. See Fig. 7 (middle and right). We encode jigsaw tabs as positive integers in binary in clockwise direction, and we encode jigsaw pockets as positive integers in binary in counterclockwise direction. Thus, two polyomino pieces fit together seamlessly precisely if the corresponding jigsaw pieces matched. To prevent pockets from conflicting with each other, we keep every 2×2 corner of the $s \times s$ square intact. Flat jigsaw edges convert to flat edges of the $s \times s$ square in the polyomino. This description concludes the conversion of jigsaw puzzles into polyomino packing puzzles.

Combining this conversion with Corollary 2, we obtain NP-hardness of the desired subfamily of polyomino packing puzzles:

Corollary 3. *It is NP-complete to decide whether n given polyomino pieces, each fitting within an $\Theta(\log n) \times \Theta(\log n)$ rectangle, can be exactly packed into a specified square box whose area equals the total area of the pieces.*

8. Closing the Loop of Reductions

Our last reduction is not for proving NP-hardness, but rather to establish a bidirectional equivalence among the puzzles we consider in this paper: edge matching, signed edge matching, jigsaw, and polyomino packing. We have already shown reductions in this order (where the case of signed edge matching and jigsaw puzzles is already bidirectional). Thus it remains to close the loop by converting polyomino packing puzzles into edge-matching puzzles:

Theorem 7. *Every polyomino packing puzzle with a box of any simple orthogonal polygon that fits inside an $x \times y$ rectangle, and with t piece types each of area at most A, has an equivalent unsigned edge-matching puzzle with an $(x+2) \times (y+2)$ box, $(x+2)(y+2)$ tiles, and $O(tA)$ colors and tile types, with $2(x+y+4)$ uniquely occurring colors.*

We have actually already seen a special case of this reduction, in Section 4, where we effectively simulated $1 \times n$ rectangle packing puzzles by edge-matching puzzles. We use the frame construction from Fig. 4, modified to force the outline of the specified simple orthogonal polygon as the box. We also unify the two interior edge colors \$ and % into a single color %. Then we create a square tile for every square in a polyomino, as shown in Fig. 8. The edges along the perimeter of a polyomino have color %. Every edge interior to a polyomino receives a unique color (occurring in a pair). Because the frame construction occupies the entire box boundary of the edge-matching puzzle, adjacent tiles belonging to a common polyomino must be placed together, rigidifying the polyomino pieces. Therefore this edge-matching puzzle has a solution precisely if the polyomino packing puzzle has a solution, concluding the reduction.

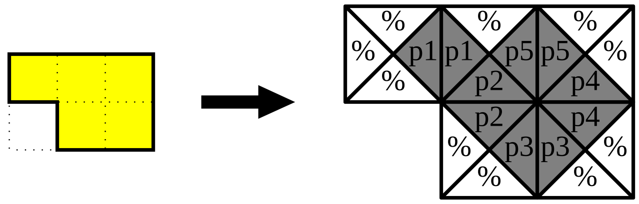

Fig. 8. Representing a polyomino of area A by A edge-matching tiles.

Acknowledgments

Thanks to Jin Akiyama for providing us with a copy of [1] which inspired Fig. 1, to Mikio Kano for citation information, and to the anonymous referees for helpful comments on the presentation of the paper.

References

1. Jin Akiyama. *Let's Enjoy Mathematics! Let's Make Mathematical Instruments by Yourself (in Japanese)*. NHK (Japan Broadcasting Corporation), 1999. Textbook for TV program for elementary-school students, episodes from July 26 to August 6.

2. Jin Akiyama and Gisaku Nakamura. An efficient dissection for a sequentially n-divisible square. In *Proceedings of the Discrete and Computational Geometry Workshop*, pages 80–89, Tokyo, Japan, 1997.

3. Jin Akiyama, Gisaku Nakamura, Akihiro Nozaki, Ken'ichi Ozawa, and Toshinori Sakai. The optimality of a certain purely recursive dissection for a sequentially n-divisible square. *Computational Geometry: Theory and Applications*, 24(1):27–39, 2003.

4. Lew Baxter. Polyomino puzzles are NP-complete. Message on Eternity Yahoo! group. http://groups.yahoo.com/group/eternity/message/3988.

5. Robert Berger. The undecidability of the domino problem. *Memoirs of the American Mathematical Society*, 66, 1966.

6. James R. Bitner and Edward M. Reingold. Backtrack programming techniques. *Communications of the ACM*, 18(11):651–656, 1975.

7. R. L. Brooks, C. A. B. Smith, A. H. Stone, and W. T. Tutte. The dissection of rectangles into squares. *Duke Mathematical Journal*, 7:312–340, 1940.

8. J. H. Conway. Mrs. Perkins's quilt. *Proceedings of the Cambridge Philosophical Society*, 60:363–368, 1964.

9. A. J. W. Duijvestijn. Simple perfect squared square of lowest order. *Journal of Combinatorial Theory, Series B*, 25(2):240–243, 1978.

10. Leah Epstein and Rob van Stee. Online square and cube packing. *Acta Informatica*, 41(9):595–606, October 2005.

11. P. Erdős and R. L. Graham. On packing squares with equal squares. *Journal of Combinatorial Theory. Series A*, 19:119–123, 1975.

12. H. Freeman and L. Gardner. Apictorial jigsaw puzzles: the computer solution of a problem in pattern recognition. *IEEE Transactions on Electronic Computers*, 13:118–127, 1964.

13. Erich Friedman. Packing unit squares in squares: A survey and new results. *The Electronic Journal of Combinatorics*, Dynamic Survey DS7, November 2005. http://www.combinatorics.org/Surveys/ds7.html.

14. Martin Gardner. Squaring the square. In *The Second Scientific American Book of Mathematical Puzzles and Diversions*, chapter 17, pages 186–209. The University of Chicago Press, 1961.

15. Martin Gardner. Mathematical games. *Scientific American*, pages 110–121, January 1977. Reprinted as "Penrose Tiling," chapter 1 of *Penrose Tiles to Trapdoor Ciphers*, published by W. H. Freeman (1989) and The Mathematical Association of America (1997).

16. Martin Gardner. Mrs. Perkins' quilt and other square-packing problems. In *Mathematical Carnival*. Vintage, New York, 1977.

17. M. R. Garey and D. S. Johnson. Complexity results for multiprocessor scheduling under resource constraints. *SIAM Journal on Computing*, 4(4):397–411, 1975.

18. Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., 1979.

19. David Goldberg, Christopher Malon, and Marshall Bern. A global approach to automatic solution of jigsaw puzzles. In *Proceedings of the 18th Annual Symposium on Computational Geometry*, pages 82–87, Barcelona, Spain, 2002.

20. Solomon W. Golomb. *Polyominoes: Puzzles, Patterns, Problems, and Packings*. Princeton University Press, 2nd edition, 1994.

21. Jacques Haubrich. *Compendium of Card Matching Puzzles*. Self-published, May 1995. Three volumes.

22. David Kempe. On the complexity of the reflections game. Unpublished manuscript, January 2003. http://www-rcf.usc.edu/~dkempe/publications/reflections.pdf.

23. Weixin Kong and Benjamin B. Kimia. On solving 2D and 3D puzzles using curve matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Hawaii, December 2001.

24. Joseph Y.-T. Leung, Tommy W. Tam, C. S. Wong, Gilbert H. Young, and Francis Y. L. Chin. Packing squares into a square. *Journal of Parallel and Distributed Computing*, 10(3):271–275, 1990.

25. Leonid A Levin. Average case complete problems. *SIAM Journal on Computing*, 15(1):285–286, 1986.

26. K. Li and K. H. Cheng. Complexity of resource allocation and job scheduling problems on partitionable mesh connected systems. In *Proceedings of the 1st Annual IEEE Symposium on Parallel and Distributed Processing*, pages 358–365, May 1989.

27. Stephen R. Mahaney. Sparse complete sets for NP: solution of a conjecture of Berman and Hartmanis. *Journal of Computer and System Sciences*, 25(2):130–143, 1982.

28. Cristopher Moore and John Michael Robson. Hard tiling problems with simple tiles. *Discrete and Computational Geometry*, 26(4):573–590, 2001.

29. Ed Pegg, Jr. The Eternity Puzzle. http://www.mathpuzzle.com/eternity.html, May 2002.

30. G. M. Radack and N. I. Badler. Jigsaw puzzle matching using a boundary-centered polar encoding. *Computer Graphics and Image Processing*, 19:1–17, 1982.

31. Alex Selby. Eternity page. http://www.archduke.org/eternity/, May 2006.

32. J. D. Skinner, II, C. A. B. Smith, and W. T. Tutte. On the dissection of rectangles into right-angled isosceles triangles. *Journal of Combinatorial Theory, Series B*, 80(2):277–319, 2000.

33. G. B. Trustrum. Mrs Perkins's quilt. *Proceedings of the Cambridge Philosophical Society*, 61:7–11, 1965.

34. W. T. Tutte. Squaring the square. *Canadian Journal of Mathematics*, 2:197–209, 1950.

35. W. T. Tutte. The quest of the perfect square. *The American Mathematical Monthly*, 72(2, part II):29–35, 1965.

36. G. N. Watson. *Messenger of Mathematics*, 48:1–22, 1918.

37. Anne D. Williams. *Jigsaw Puzzles: An Illustrated History and Price Guide*. Wallace-Homestead Book Co, Radnor, PA, 1990.

38. Anne D. Williams. *The Jigsaw Puzzle: Piecing Together a History*. Berkley Books, New York, 2004.

39. Thomas Wolf. The 70^2 puzzle. http://home.tiscalinet.ch/t_wolf/tw/misc/squares.html, May 2003.

40. H. Wolfson, A. Kalvin, E. Schonberg, and Y. Lambdan. Solving jigsaw puzzles by computer. *Annals of Operation Research*, 12(1–4):51–64, February 1988.

Received: May 29, 2006
Final version received: January 31, 2007