Journal of Information Processing Vol.0 No.0 1–13 (??? 1992)

[DOI: 10.2197/ipsjjip.0.1]

Regular Paper

Total Tetris: Tetris with Monominoes, Dominoes, Trominoes, Pentominoes, ...

Erik D. Demaine[1,a)] Martin L. Demaine[1,b)] Sarah Eisenstat[1,c)] Adam Hesterberg[1,d)] Andrea Lincoln[1,e)] Jayson Lynch[1,f)] Y. William Yu[1,g)]

Received: January 1, 0000, Accepted: xx xx, xxxx

Abstract: We consider variations on the classic video game *Tetris* where pieces are k-ominoes instead of the usual tetrominoes ($k = 4$), as popularized by the video games *ntris* and *Pentris*. We prove that it is NP-complete to survive or clear a given initial board with a given sequence of pieces for each $k \geq 5$, complementing the previous NP-completeness result for $k = 4$. More surprisingly, we show that board clearing is NP-complete for $k = 3$; and if pieces may not be rotated, then clearing is NP-complete for $k = 2$ and survival is NP-complete for $k = 3$. All of these problems can be solved in polynomial time for $k = 1$.

Keywords: NP-complete; complexity; puzzles; algorithms

1. Introduction

Tetris is one of the most famous puzzle computer games, originally created in 1984 and released in the west on the IBM PC in 1987,[*1] and substantially popularized by being bundled with every Nintendo Game Boy (except in Japan) [7]. Today, Tetris and its many clones are available to play on almost every platform; the official Tetris mobile game sold over 425 million copies by 2014, eclipsing even the 35 million Game Boy copies [8].

The popularity of Tetris has led to many variations, both official and unofficial, with various changes to the rules. Here we consider a theme introduced by Hunter Freyer's *Pentris* [3], where pieces are pentominoes instead of tetrominoes. Later, Shaunak Kishore's *ntris* [6] generalized to pieces that are k-ominoes (made of k unit squares joined edge to edge).

Our results. In 2004, Breukelaar et al. [1] proved that Tetris is NP-complete for the original tetromino pieces; here we generalize this result to k-omino pieces. More precisely, Breukelaar et al. and we analyze an *offline* version of Tetris, where the board has a given configuration of occupied squares (resulting from past play, or a complex initial board) and the pieces come from a given sequence of n pieces, and the goals are to survive all the pieces (avoid stacking any piece too high), clear the entire board (remove all occupied squares), or maximize score (according to various measures). While the interesting score functions were clearer for tetrominoes, this aspect has many possible generalizations to k-ominoes, so we focus on the first two goals.

Table 1 summarizes our (and past) results for Tetris with k-ominoes for the goals of clearing and survival. In addition to the standard rules where pieces can be rotated and translated left/right/down, we consider a variant that forbids rotations. This variation is particularly interesting for dominoes ($k = 2$), where we can show NP-completeness for clearing, while we conjecture polynomial time with rotation allowed. In total, our results categorize the computational complexity of most variants of Tetris with k-ominoes for most k.

Tetris with monominoes ($k = 1$) is easy (Section 3): we can always survive (by placing each piece into any blank space), and clearing is equivalent to the number of falling pieces equaling the number of blank spaces in nonempty rows plus a nonnegative integer multiple of the board width. Section 4 analyzes dominoes ($k = 2$), proving NP-completeness for clearing without rotation and proving a lemma about survival with rotation. Section 5 analyzes trominoes ($k = 3$), proving NP-completeness for clearing with and without rotation, and for survival without rotation. Section 6 analyzes large k-ominoes (any $k > 4$), proving NP-completeness for all four variants. All of these problems are trivially in NP with a certificate of the sequence of piece placements. We discuss the remaining open problems in Section 8.

2. Background

Tetris is a video game that is played on a square grid with all the possible tetrominoes. In Tetris a tetromino piece appears at the top of the screen and periodically moves down in the grid. While the block is falling, the player may move the block horizontally or rotate the block by 90°. If the piece would otherwise move down into an already occupied square on the board, it instead stops moving and becomes part of the inactive block. If the piece stops above a certain height, the player loses; otherwise, a new piece

[1] Massachusetts Institute of Technology, Cambridge, MA 02139, USA
[a)] edemaine@mit.edu
[b)] mdemaine@mit.edu
[c)] seisenst@mit.edu
[d)] achester@mit.edu
[e)] andreali@mit.edu
[f)] jaysonl@mit.edu
[g)] ywy@mit.edu

[*1] Tetris was one of the very first PC games played by the first two authors.

	With Rotation	
	Clearing	Survival
$k = 1$	P (§3)	P (§3)
$k = 2$	Open	Open
$k = 3$	NP-complete (§5)	Open
$k = 4$	NP-complete [1]	NP-complete [1]
$k > 4$	NP-complete (§6)	NP-complete (§6)

	No Rotation	
	Clearing	Survival
$k = 1$	P (§3)	P (§3)
$k = 2$	NP-complete (§4)	Open
$k = 3$	NP-complete (§5)	NP-complete (§5)
$k = 4$	NP-complete [1]	NP-complete [1]
$k > 4$	NP-complete (§6)	NP-complete (§6)

Table 1: Complexity results for Tetris with k-ominoes, with or without rotation. New results are marked with a section number (§).

appears at the top. If a row is completely filled by inactive blocks, then all of those squares are emptied and all squares above that row move down by 1.

2.1 Rules for *k*-omino Tetris

The formal rules for Tetris are laid out in the paper *Tetris is hard, even to approximate* [1]. We allow arbitrary board sizes and starting states, and the board starting state is part of the problem specification. Rows cannot be filled entirely in the start state. We also do not allow rows to "float" above completely clear rows. We generally require that the initial board state is constructible, i.e., could be reached from an empty board by a sequence of moves in the game. Constructibility has a general characterization [5], but we give explicit constructions to make this paper self-contained.

Piece sets. For $k < 4$, we will present separate proofs for Tetris hardness using the piece sets of either the monomino (1-omino), domino (2-omino), and tromino (3-omino). For $k > 4$, we use the same generalizable proof for all such k using only a small subset of the larger polyominoes, described in section 6.

Survival vs. completely clear board. In this paper we consider two objectives in Tetris: clearing the board and survival. Clearing the board is the goal of having all rows clear at the moment when all input pieces have been used. Survival is the goal of not placing any of the pieces above the upper limit of the board.

Rotation vs. no rotation. In the standard Tetris game, the player can rotate pieces by any integer multiple of 90°. We consider both the case where rotation is and is not allowed. When rotation is not allowed, the pieces must stay in their specified orientations.

2.2 Problem to Reduce from

The original proof that Tetris is hard [1] reduces from the problem of 3-Partition [4]. All of the proofs in this paper use a similar construction, in particular reducing from the same problem:

Problem 1 (3-Partition). *Given a multiset $A = \{a_1, a_2, \ldots, a_{3m}\}$ of positive integers such that $t/4 < a_i < t/2$ for each i, where $t = \frac{1}{m}\sum_{i=1}^{3m} a_i$, determine whether there is a partition of A into m groups $D_1, \ldots, D_m$ each of size 3 and having sum $\sum_{x \in D_j} x = t$.*

All of our hardness proofs share a common core. We create a starting board state consisting of m vertical buckets of equal height ($\Theta(t)$ to incorporate some scaling), forcing the player to partition their input sequence into the buckets. We create a sequence of Tetris pieces S_i to represent each input number $a_i \in A$, which forces all of the blocks representing a given a_i to go into one bucket. More precisely, the overall piece sequence is the concatenation $S_1, S_2, \ldots, S_{3m}$, and each piece sequence S_i decomposes further into a constant-length ***priming sequence*** (forcing the choice of a single bin), a ***filling sequence*** of length $\Theta(a_i)$ (representing the number), and a constant-length ***closing sequence*** (undoing the priming sequence).

3. Monominoes

Unsurprisingly, Tetris with monominoes is easy. Rotation does nothing to the 1×1 block. If there is a hole, these blocks fit. We include these proofs for completeness.

Theorem 3.1. *Survival with monominoes is in P, and always a "yes" instance.*

Proof. We claim that survival with monominoes is always possible. There must always be an empty square in the top row, because if the top row were completely filled, it would clear. So, every monomino can be placed in any available location. Therefore, we can always return true, in $O(1)$ time. □

Theorem 3.2. *Clearing with monominoes is in P.*

Proof. Let h be the number of holes in partially filled rows (easily counted in linear time), and let w be the width of the board (size of a row). Then m monominoes can clear the board if and only if $m \geq h$ and $(m - h) \bmod s = 0$. Both conditions are obviously necessary. The strategy that proves sufficiency is to repeatedly place a piece in any hole in any partially filled row (whichever is accessible), except when the board is already cleared, in which case the piece can go in any square in the bottom row. □

4. Dominoes

In this section, we consider 2-tris with domino pieces both without and with rotation. We show that the former is NP-complete, and prove a lemma about the latter.

4.1 Clearing without Rotation

Theorem 4.1. *Clearing the board in 2-tris without rotation is NP-complete.*

Proof. We reduce from 3-Partition; refer to Figure 1.

The board has width $5m$ and height $3m + 2t$. Each of the m buckets consists of five adjacent columns, where the first and last columns are filled and the middle column is empty. In the bottom $2t$ rows, the second and fourth columns are filled. In the top $3m$ rows, we have a *blocking structure* in which the second and fourth columns alternate between filled and empty, with opposite start parity.

To represent a_i, we start with a priming sequence of $m - 1$ horizontal pieces, which will block all but one of the buckets at the top of the blocking structure. Then we have a filling sequence of a_i vertical pieces, which will all go in the remaining bucket. Finally, we have a closing sequence of 2 horizontal pieces, which unprime the bucket by clearing the top row of the blocking structure.

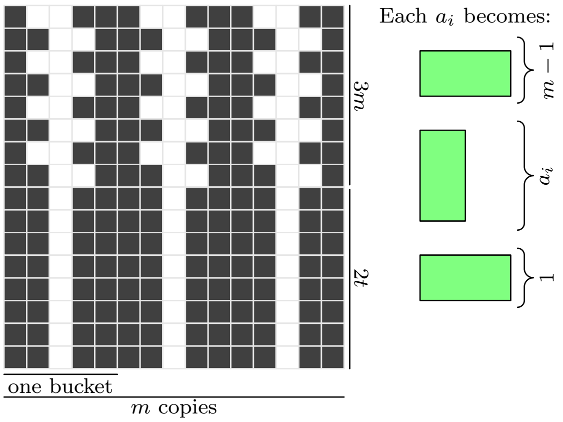


Fig. 1: Reduction from 3-Partition to clearing 2-tris without rotation.

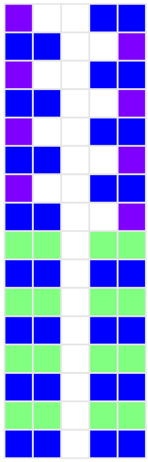


Fig. 2: Constructibility of the board in Figure 1.

The area of holes in the board is equal to the area of the pieces we give to the player. Thus, in order to clear the board, every piece must be placed within the (original) holes. Similarly, the bottom part of the board can be filled only with vertical pieces, and its area equals the total area of the vertical pieces. Thus, in order to clear the board, every vertical piece must be placed in the bottom part of the board.

Consider the piece sequence S_i representing a_i. The priming sequence of $m-1$ horizontal pieces must be placed in the top nonempty row, and thus must block off $m-1$ buckets. One bucket remains unblocked, so the filling sequence of a_i vertical pieces must go in that bucket, and as argued above, in the bottom of that bucket. The closing sequence of 1 horizontal piece must go in the top row of the one remaining bucket, which clears that row and opens back up all of the buckets.

If there is a 3-partition, then we can clear the 2-tris board as follows. For each group $\{a_i, a_j, a_k\}$, we put all of their $a_i + a_j + a_k = t$ associated vertical pieces into the same bin (and no other vertical pieces in that bin), filling the $2t$ rows of this bucket, while also clearing three of the upper rows with the $3m$ associated horizontal pieces. Once all buckets are so filled, the entire board clears.

If we attempt an invalid 3-partition where some groups do not sum to t, then by an area argument, some bucket must have fewer than t vertical pieces placed into it, leaving that bucket unfilled and hence some of its rows uncleared.

Figure 2 shows how to construct the board with dominoes. The bottom $2t$ rows are straightforward. Every row in the top blocking structure has a desired singleton (in purple). We construct the singletons in each row by placing a vertical piece on top of it, then adding horizontal pieces to fill the rest of the row. Consecutive singletons on a common row (from multiple buckets) have four spaces between them, covered by two horizontal pieces. Once the row is filled, the horizontal pieces clear, leaving just the singletons we desired. □

4.2 Toward Survival with Rotation

We conjecture that surviving dominoes with rotation is in P. Here we show a lemma toward this goal: clearing a single row leads to survival.

Lemma 4.2. *If the top row of the board is clear, then the player can survive arbitrarily long in 2-tris with rotation.*

Proof. Rotate all dominoes to be vertical. Greedily place a piece in any column with an empty square in both of the top two rows. Throughout, we maintain the invariant that the filled spaces of the top row are a subset of the filled spaces of the second row. Eventually, the second row (and possibly the first row) of each column will become filled, at which point the second row clears, and the first row becomes the second row, returning to the case where the top row is clear. □

To show that this problem is in P using this method, we must additionally show that

- determining whether we can clear the top row is in P, and
- deciding survival in cases where the top row cannot be cleared is in P.

Note that clearing the top row is equivalent to clearing any row, since clearing any row causes rows to shift down until the top row is empty.

5. Trominoes

In this section, we show NP-completeness of clearing 3-tris with rotation, and of survival and clearing in 3-tris without rotation.

5.1 Clearing with Rotation

Theorem 5.1. *Clearing the board in 3-tris with rotation is NP-complete.*

Proof. We reduce from 3-Partition; refer to Figure 3. The proof is structurally similar to that of Theorem 4.1. Again, each bucket has width 5, for a total board width of $5m$; the first and last columns of each bucket are filled while the middle column is empty. The bottom $3t$ rows (instead of $2t$ rows as in the corresponding 2-tris proof) are filled in the second and fourth columns. The top $6m$ rows (instead of $3m$ rows) form a blocking structure, with the periodic pattern shown in the figure. For each a_i, the priming sequence is $m-1$ Ls (instead of horizontal dominos), the filling sequence is a_i straight trominos (instead of vertical dominoes), and the closing sequence is an L. These three sequences serve the same role as before: blocking all but one bucket, filling that bucket with the number, and then unblocking all buckets. Figure 4 shows the intended behavior.

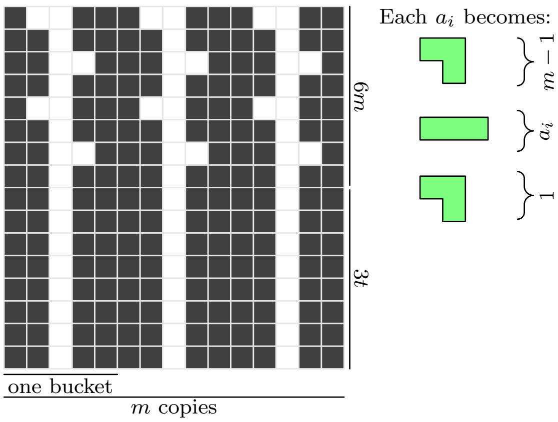


Fig. 3: Reduction from 3-Partition to clearing 3-tris with rotation.

As before, the area of holes is equal to the area of the pieces, so to clear the board, all pieces must be placed within the (original) holes. Similarly, the bottom part of the board can be filled only by straight tromino pieces, so by an area argument, all tromino pieces must go there. Therefore, as in Figure 4, the priming sequence must block $m - 1$ of the buckets, and the filling sequence and closing sequence must go in the remaining bucket, filling some of the bottom of that bucket and filling then clearing two rows. Hence, as before, there is a valid clearing if and only if there is a 3-partition.

If m is even, then the board is constructible by tiling; see Figure 5. □

5.2 Without Rotation

When no rotation is allowed, we can show hardness of both clearing and survival.

Theorem 5.2. *Surviving or clearing* 3*-tris without rotation is NP-complete.*

Proof. We reduce from 3-Partition; refer to Figure 6.

The board has width $4m + 1$ and height $t + 8$. The columns divide into m buckets followed by a single empty column. Each bucket consists of four columns, the rightmost of which is filled and the rest of which are empty except possibly in the bottommost two rows. All buckets start ***unprimed*** meaning that their first three columns descend as a staircase with increments of -1, i.e., with column heights of $k + 2, k + 1, k$ for some k.

To represent a_i, we start with a priming sequence of one L tromino pointing up and right, followed by a filling sequence of a_i horizontal straight trominoes, followed by a closing sequence of one L tromino pointing down and left. The priming sequence can be placed in two ways into an unprimed bucket, as shown in Figure 7. (Unlike the previous two hardness proofs which blocked all but the desired bucket, this priming sequence reshapes the desired bucket.) If placed to the right, the top surface of the bucket becomes flat, with column heights k, k, k, which we call ***primed***. In any bucket that is not primed, placing a horizontal piece will make an internal hole. In a primed bucket, placing the horizontal pieces in the filling sequence creates no internal holes and preserves the invariant that the bucket is primed. The closing sequence can also be placed in two ways in a bucket; on the left, it transforms a primed bucket back into an unprimed bucket. If placed on the right, later Ls and horizontal pieces will be forced to create internal holes.

After the $S_1, S_2, \ldots, S_{3m}$ piece sequences representing the numbers $a_1, a_2, \ldots, a_{3m}$, we have a final piece sequence of m L trominoes pointing up and right followed by $(t + 8)/3$ vertical straight trominoes. (Here we assume that $t + 8$ is divisible by 3.) Only these vertical straight trominoes can fit into the rightmost column, so no lines can be cleared until these pieces come. The area of all other empty space in the initial board equals the total area of the non-vertical pieces. Therefore, if the player makes any internal holes, they will run out of space to place pieces before getting to the vertical pieces. This fact forces the behavior of each a_i sequence to convert an unprimed bucket to primed, fill that bucket, and then convert that primed bucket back to unprimed. The buckets will not overflow if and only if we follow a 3-partition. In this case, the final m L trominoes finish filling the (unprimed) buckets, and the vertical trominoes fill the rightmost column, clearing the board.

The board is constructible by tiling as long as its height $t + 8$ is divisible by 3, which we already assumed. □

6. Clearing with Larger Polyominoes

In this section, we consider the case of Tetris when the pieces are generalized to polyominoes consisting of $n > 4$ unit squares. We adapt some of the techniques used to show the hardness of Tetris to show the hardness of the clearing variant of n-tris for any $n > 4$. In Section 6.1, we define some useful terms, explain how to set up the board, and give a general proof showing that n-tris is NP-complete if certain piece sequences can be constructed. In Section 6.2, we give a way to construct these piece sequences for Pentris (5-tris), the variant of Tetris with pentominoes. In Section 6.3, we explain how to generalize this construction to handle n-ominoes for all $n > 5$.

6.1 General *n*-tris Properties

For each problem, we reduce from the 3-Partition problem. For each proof, we set the width w of the n-tris board to $5m + 2$. We split the height of the board into two sections: a completely blank space at the top for transforming and rotating pieces, and a height-h space at the bottom which has each row partly filled. The value of h will be different for each of the proofs, but will always be a multiple of n. The sequence of pieces we give will have area exactly equal to the number of empty cells in the lower h rows of the board, so if the player wishes to clear all h rows with the given sequence of pieces, all pieces must be placed entirely inside the bottom h rows.

We number the h rows of the lower section of the n-tris board from bottom to top, so that the index of the lowest row is 1 and the index of the highest row is h. We say that a column is ***packed*** if all cells below the highest filled cell are also filled. In the initial board we construct, we want all columns to be packed. The ***height*** of a column is the row of the highest filled cell in that column, or 0 if the whole column is empty.

The last two columns of the board we construct are special. The

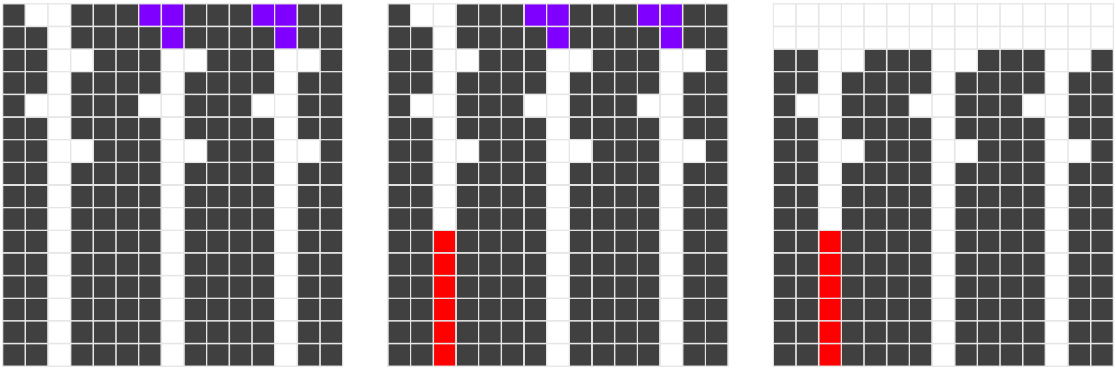


Fig. 4: Placing $a_i = 2$ in Figure 3.

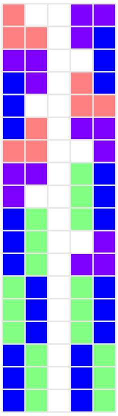


Fig. 5: Constructibility of the board in Figure 3.

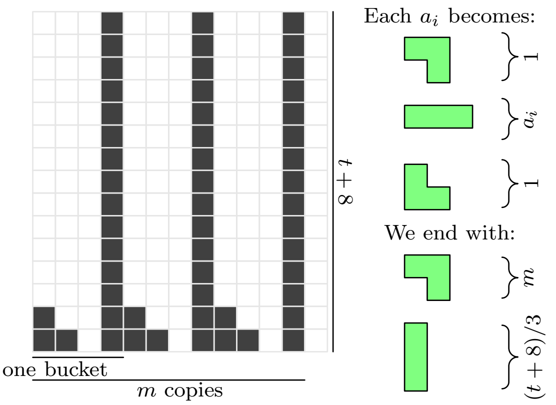


Fig. 6: Reduction from 3-Partition to surviving or clearing 3-tris without rotation.

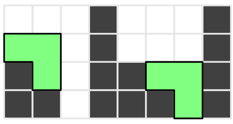


Fig. 7: The two possible ways to place the priming sequence in an unprimed bucket in Figure 6.

rightmost column, $5m+2$, will be completely empty (with a height of 0). The column $5m+1$ will be completely filled (packed to a height of h). Thus, in order to fill in any cells in the rightmost column, we must have a $1 \times n$ piece (as any other piece, when inserted into that column, would extend above row h). This makes it impossible to clear any rows until we get a $1 \times n$ piece to fit into the rightmost column. The sequence of pieces we construct takes advantage of this fact. In particular, we construct a sequence of pieces (none of which are the $1 \times n$ piece) with area exactly equal to the number of empty cells in the rest of the board, and then follow it up with (h/n) copies of the $1 \times n$ piece, just enough to fill the last column. Thus, if the player leaves any holes when dropping the initial sequence of pieces, they will be forced to place a piece above the top of the board before they can clear any rows.

A ***bucket*** consists of five adjacent columns. The first and last columns of the bucket are packed to height h, so that any pieces have to be placed in the center columns between the sides of the bucket. The ***shape*** of a bucket is a function of the heights h_1, h_2, h_3 of its three center columns:

$$\langle h_1 - \min\{h_1, h_2, h_3\}, h_2 - \min\{h_1, h_2, h_3\}, h_3 - \min\{h_1, h_2, h_3\}\rangle.$$

The ***depth*** of a bucket is the difference between the height h of the board and the minimum height $\min\{h_1, h_2, h_3\}$ of any of the central columns.

The original Tetris paper [1] defines a ***choke point*** in a bucket to be a row where all but one cell in the bucket is filled. If there is an empty cell in the bucket that is lower than the choke point but in a different column, then the only way to fill that cell is by sliding in the only piece that fits through the choke point: a $1 \times n$ piece. Because we do not use $1 \times n$ pieces in the first phase of our n-tris game, we know that if there is such a cell, it must remain empty until we reach the last h/n pieces in our sequence. However, that means that we have at least $h+1$ cells being filled by h/n pieces of total area h, and therefore at least one cell must be empty at the end of the sequence (which, in turn, means that at least one cell above row h must be filled). Thus, these two rules can be used to determine whether a particular piece placement is ***valid*** — that is, whether it will allow all cells in the bucket to be filled before the first $1 \times n$ bar:

(1) No piece can be placed so that the empty space in a bucket becomes disconnected

(2) No piece can be placed to cause a choke point with an empty cell below the choke point in a different column.

For $i \in \{1, \ldots, m\}$, we construct a bucket in columns $5i-4$ through $5i$. All buckets should initially start with the same shape, which we will call the ***unprimed shape*** s_U. For each of the variants of n-tris, we must construct four piece sequences, each of which alters a bucket in different ways. The first sequence can only be used to change the shape of a single bucket into a ***primed shape*** s_P. Formally, we define the priming sequence as follows:

Definition A ***(s_U, s_P)-priming sequence*** is a non-empty sequence of n-ominoes with the following properties:

(1) The sequence does not contain any $1 \times n$ pieces.

(2) Only the first piece can be placed into a bucket with shape s_U.

(3) There is exactly one valid way to place the pieces in a bucket with shape s_U, and the result will be a packed bucket with shape s_P.

Once the shape of a bucket has been changed to s_P, we want to fill the bucket with a number of pieces proportional to a_i, the number from the 3-Partition instance we are trying to represent. Thus, we want a sequence that does not change the shape of the bucket (so that it can be repeated arbitrarily many times), and cannot fit into an unprimed bucket:

Definition A ***(s_U, s_P)-filling sequence*** is a non-empty sequence of n-ominoes with the following properties:

(1) The sequence does not contain any $1 \times n$ pieces.

(2) There is no valid way to place any of the pieces into a bucket with shape s_U.

(3) There is exactly one valid way to place the pieces in a bucket of shape s_P, and the result will be a packed bucket with shape s_P.

Finally, once the bucket has been filled with a_i copies of the filling sequence, we want to close the bucket so that the shape becomes unprimed again:

Definition A ***(s_U, s_P)-closing sequence*** is a non-empty sequence of n-ominoes with the following properties:

(1) The sequence does not contain any $1 \times n$ pieces.

(2) There is no valid way to place any of the pieces into a bucket with shape s_U.

(3) There is exactly one valid way to place the pieces in bucket with shape s_P, and the result will be a packed bucket with shape s_U.

These three sequences allow us to force a long sequence of pieces to be placed in one bucket, which helps us represent the 3-Partition instance as a packing problem. The only other thing we need is a sequence of pieces that can be used to flatten the top of each bucket after they have been fully packed:

Definition A ***s_U-flattening sequence*** is a $(s_U, \langle 0, 0, 0\rangle)$-priming sequence.

Theorem 6.1. *For any $n > 4$, if there exists a pair of bucket shapes s_U, s_P such that the set of n-ominoes can be used to construct a (s_U, s_P)-priming sequence, a (s_U, s_P)-filling sequence, a (s_U, s_P)-closing sequence, and an s_U-flattening sequence, then it is NP-complete to clear the board in n-tris.*

Proof. Let P be the (s_U, s_P)-priming sequence, let F be the (s_U, s_P)-filling sequence, let C be the (s_U, s_P)-closing sequence, and let L be the s_U-flattening sequence. Let h_P be the difference in the height of the center column of an unprimed bucket after adding the sequence P. Let h_F be the difference in the height of the center column of a s_P-shaped bucket after adding the sequence F. Let h_C be the difference in the height of the center column of a s_P-shaped bucket after adding the sequence C. Let h_L be the difference in the height of the center column of an unprimed bucket after adding the sequence L. Note that because each sequence contains at least one piece, and the sequence does not contain the $1 \times n$ piece (so all pieces must fill at least one cell in the center column), we know that $h_P, h_F, h_C, h_L \geq 1$.

For each a_i, the associated sequence $S(a_i)$ is defined as follows:

$$S(a_i) = P \circ \underbrace{F \circ F \circ \ldots \circ F}_{(h_L+1)\cdot a_i \text{ times}} \circ C.$$

We set up the buckets so that initially, the height of the center column is:

$$h - (h_L + 3(h_P + h_C) + t \cdot h_F \cdot (h_L + 1)).$$

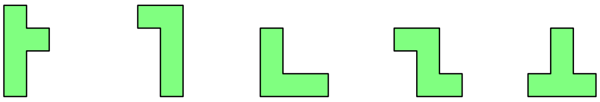
Let I be the $1 \times n$ piece. Then we create the following sequence of pieces:

$$S(a_1) \circ \ldots \circ S(a_{3m}) \circ \underbrace{L \circ \ldots \circ L}_{m \text{ times}} \circ \underbrace{I \circ \ldots \circ I}_{h/n \text{ times}}.$$

We wish to show that this sequence can be placed into a $h \times (5m+2)$ board, with m buckets of shape s_U and a single unfilled column at index $5m + 2$, if and only if there is a solution to the original 3-Partition instance.

Consider the placement of the pieces in $S(a_i)$. By definition, only the very first piece of $S(a_i)$ can be placed in an unprimed bucket without producing a hole or a choke-point. Hence, all the pieces of $S(a_i)$ must be placed in the same bucket. Furthermore, we know that there is exactly one way to place the sequence P in an unprimed bucket with shape s_U, and the result will make the shape of the bucket s_P. We also know that each sequence F can be arranged in exactly one way in a bucket of shape s_P, and the result will be a bucket of shape s_P. Finally, we know that the sequence C can be arranged in exactly one way in a bucket of shape s_P, and the result will be a bucket of shape s_U. Thus, once the first piece has been placed, the entire sequence $S(a_i)$ has exactly one valid configuration, and the resulting shape of the bucket will be s_U. This will increase the height of the center column of the bucket by $h_P + (h_L + 1) \cdot a_i \cdot h_F + h_C$.

Suppose that have a solution to the n-tris instance that places k sequences $S(a_{i_1}), \ldots, S(a_{i_k})$ inside the same bucket. Then the height of the center column of that bucket will become:



(**a**) Little T. (**b**) Long L. (**c**) Short L. (**d**) Z. (**e**) Big T.

Fig. 8: The pieces used in the Pentris hardness proof.

$$
\begin{aligned}
h-&(h_L + 3(h_P + h_C) + t \cdot h_F \cdot (h_L + 1)) \\
&+ \sum_{j=1}^{k} (h_P + h_C) + (h_L + 1) \cdot h_F \cdot a_{i_j} \\
= h-&h_L - 3(h_P + h_C) - t \cdot h_F \cdot (h_L + 1) \\
&+ k(h_P + h_C) + (h_L + 1) \cdot h_F \cdot \sum_{j=1}^{k} a_{i_j} \\
= h-&h_L + (k-3)(h_P + h_C) \\
&+ \left(\left(\sum_{j=1}^{k} a_{i_j} \right) - t \right) \cdot h_F \cdot (h_L + 1).
\end{aligned}
$$

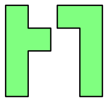
Suppose first that $k \geq 4$. Then, because each $a_i > t/4$, we know that the sum of the a_{i_j}s must be strictly greater than t, so the resulting height will be at least $h - h_L + (h_P + h_C) + h_F \cdot (h_L + 1) \geq h - h_L + h_L + 1 = h + 1$, and the player will lose. Next suppose that $k = 3$ and the sum of the a_{i_j}s is strictly greater than t. Then the resulting height will be at least $h - h_L + h_F \cdot (h_L + 1) \geq h + 1$, which again introduces a contradiction. So we know that each bucket can only fit at most three sequences $S(a_i)$, and that the numbers used to generate those three sequences can sum to at most t. Because there are $3m$ sequences and m buckets, this means that each bucket must contain exactly three sequences. Furthermore, we know that $t = \frac{1}{m} \sum a_i$, so if each partition sums to at most t, each partition must sum to exactly t. Hence, we have a partition of A into m sets of size 3 such that each set sums to t, which is precisely what we wanted.

Suppose instead that we have a solution to the 3-Partition instance, and want to solve the corresponding n-tris problem. Let $B_1, \ldots, B_m$ be the solution to the 3-Partition instance, and let $K_1, \ldots, K_m$ be a partition of the indices $\{1, \ldots, 3m\}$ into groups so that $B_i = \{a_j \mid j \in K_i\}$. Then for each $j \in \{1, \ldots, 3m\}$, we place the sequence $S(a_j)$ into the ith bucket, where i is selected so that $j \in K_i$. There is exactly one way to place each sequence once the bucket has been chosen, so this uniquely determines how to place each sequence $S(a_i)$. Note that, after this is done, the height of the center column of each bucket will be exactly $h - h_L$. Thus, we can use the m copies of L to completely fill in the remaining cells in each of the buckets. Finally, once each of the buckets has been completely filled, we can drop the h/n copies of the $1 \times n$ piece into the last column, and clear the entire board. □

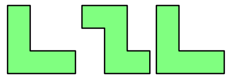
6.2 Pentris

For the Pentris hardness proof, we use the unprimed shape $s_U = \langle 0, 1, 0 \rangle$, and the primed shape $s_P = \langle 0, 0, 0 \rangle$. Note that this means that the (s_U, s_P)-priming sequence is the same as the s_U-flattening sequence, so it is sufficient to show the existence of only three sequences: priming, filling, and closing.

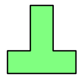
We use a total of five different pieces, depicted in Figure 8. Figure 8(a) depicts the piece used to start the priming phase, which



(**a**) The priming sequence.



(**b**) The filler sequence.



(**c**) The closing sequence.

Fig. 9: The sequences of pieces used for the Pentris proof.

we call the ***little T***. Figure 8(b) depicts the piece used to finish the priming phase, which we call the ***long L***. Figure 8(c) is the ***short L*** used as both the first and last piece in the filling sequence. Figure 8(d) is the ***Z piece*** used as the second piece in the filler phase. Figure 8(e) is the ***big T*** used in the closing phase. None of these pieces are the 1×5 piece, so any sequence constructed using these pieces will satisfy Property 1 of the definitions of the priming, filling, and closing sequences.

Figure 9 shows the three sequences used for the Pentris proof. In order to satisfy Property 2 of each of the definitions, we must show that only the little T piece can be placed in an unprimed bucket. Figure 10 shows what happens when any other piece is used in an unprimed bucket. All possible orientations and positions lead to unfillable spaces, which will ultimately cause the player to lose. Hence, Property 2 is satisfied for all three sequences.

Next, we must show that Property 3 holds for all three sequences. We begin with the priming sequence. Figure 11 shows that there is only one possible orientation and position for the little T piece, and Figure 12 shows that, given the location for the first priming piece, there is only one way to place the subsequent long L piece. Hence, the positions of the two pieces in the priming sequence are fixed, and it is clear to see that the bucket will have the desired $s_P = \langle 0, 0, 0 \rangle$ shape as a result of placing them. This shows that these two pieces form a (s_U, s_P)-priming sequence (as well as a s_U-flattening sequence, because $s_P = \langle 0, 0, 0 \rangle$).

Unfortunately, the filler sequence is not similarly constrained. As Figure 13 shows, There are two valid positions for the short L piece in a primed bucket with shape $\langle 0, 0, 0 \rangle$. However, as Figure 14 shows, once the second piece in the filler sequence has been placed, there is only one configuration of the first two pieces that will not create holes. When we add the second short L piece to the initial pieces in the filler sequence, as in Figure 15, the result is a sequence of three pieces that has exactly one valid configuration in a bucket with shape $\langle 0, 0, 0 \rangle$, and that configuration will not change the shape of the bucket. Thus, this is a valid (s_U, s_P)-filling sequence.

Finally, we consider the closing sequence, which consists only of the big T piece. Figure 16 shows that there is only one valid configuration for the piece in a primed bucket, and that the shape of the bucket after the piece has been placed will be $s_U = \langle 0, 2, 0 \rangle$, just as we wanted.

In this section, we have shown that there exists a valid (s_U, s_P)-priming sequence (and therefore a valid s_U-flattening sequence), a valid (s_U, s_P)-filling sequence, and a valid (s_U, s_P)-closing sequence. Hence, by Theorem 6.1, we may conclude that:

Theorem 6.2. *Clearing the board in Pentris is NP-complete.*

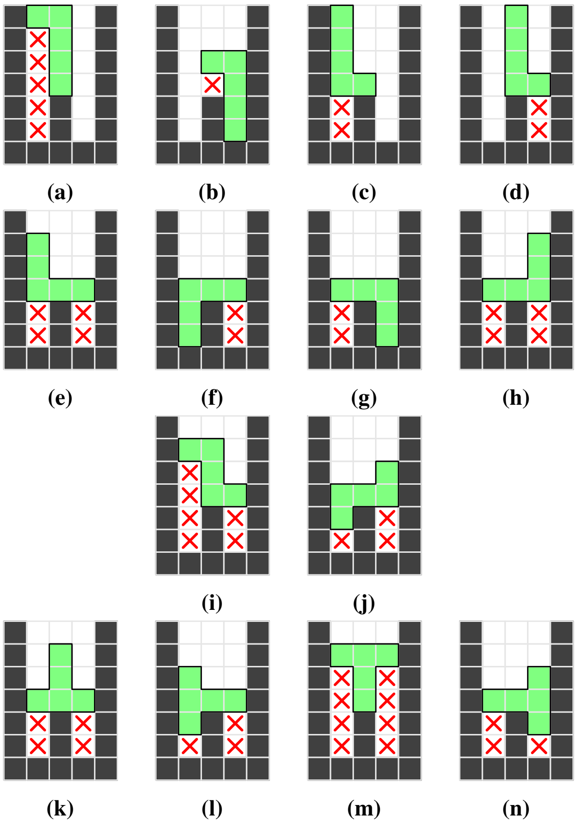


Fig. 10: Figures (a) through (d) show what happens if the long L piece (the second priming piece) is placed in an unprimed bucket. Figures (e) through (h) show what happens if the short L piece (the first and last filler piece) is used in an unprimed bucket. Figures (i) through (j) show what happens if the Z piece (the second filler piece) is used in an unprimed bucket. Figures (k) through (n) show what happens if the big T piece (the closing piece) is used in an unprimed bucket. Note that all possible configurations will result in holes or choke points, meaning the player will not be able to win.

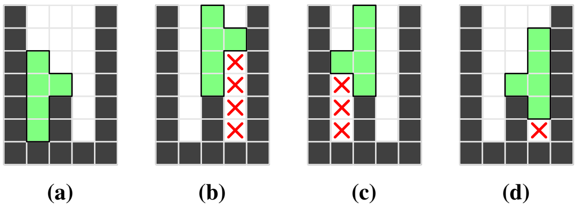


Fig. 11: Possible positions for the little T piece (the first priming piece) in an unprimed bucket. Clearly, only one placement is valid.

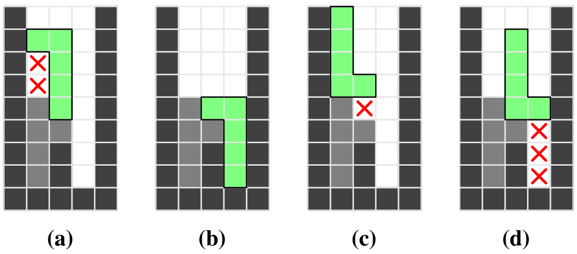


Fig. 12: Possible positions for the second piece in the priming sequence.

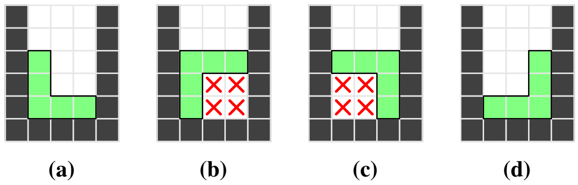


Fig. 13: Possible positions for the short L in a primed bucket.

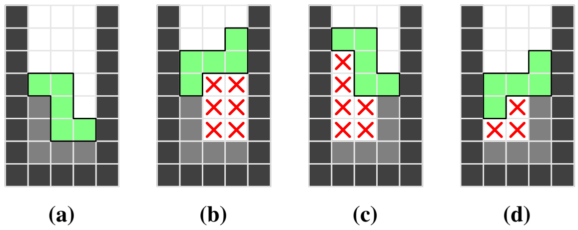


Fig. 14: Possible positions for the Z piece (the second piece in the filler sequence), given the two valid positions for the first piece in the filler sequence.

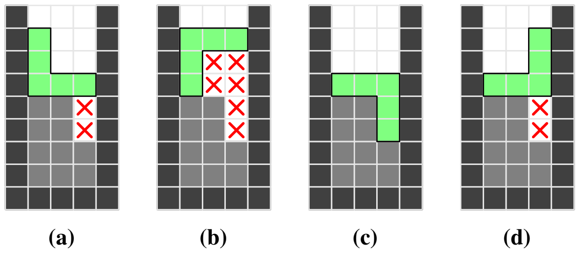


Fig. 15: Possible positions for the third piece in the filler sequence.

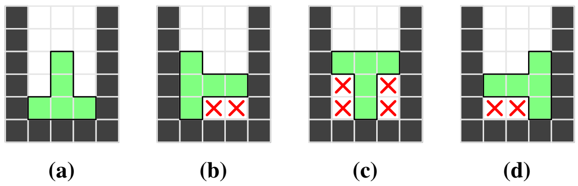


Fig. 16: Possible positions for the big T piece in a primed bucket.

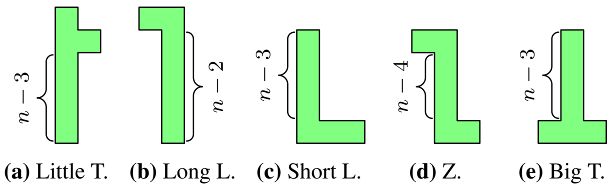
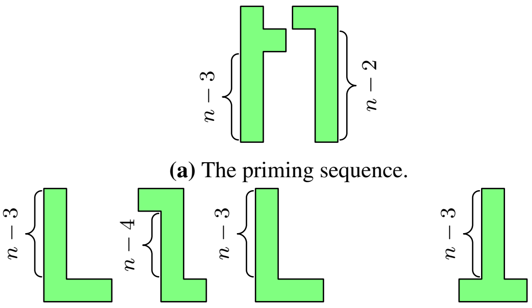


Fig. 17: The pieces used in the n-tris hardness proof for $n > 5$. Note the number of rows in each piece is at least 4 for all $n \geq 6$, so each piece has at most two possible orientations to fit in a bucket. The Z piece (d) is the same after being rotated 180°, so it has only one possible orientation.

6.3 n-tris for $n > 5$

We can generalize the Pentris proof to n-tris for any $n > 5$ by extending each of the pieces vertically, as shown in Figure 17. For this case, we set the unprimed bucket shape $s_U = \langle 0, n-3, 0 \rangle$, and we set the primed bucket shape $s_P = \langle 0, 0, 0 \rangle$ (so that, once again, the (s_U, s_P)-priming sequence is also a s_U-flattening sequence).

We use the same sequences as the Pentris proof (see Figure 18: the priming sequence uses the little T piece followed by the long L piece, the filler sequence has a Z piece sandwiched between two



(a) The priming sequence.

(b) The filler sequence. **(c)** The closing sequence.

Fig. 18: The sequences of pieces used for the n-tris proof.

short L pieces, and the closing sequence consists of a single big T piece. Clearly, none of these sequences contains a $1 \times n$ piece, so all of them satisfy Property 1 of their respective sequence definitions. Figure 19 shows what happens when each of the pieces (other than the little T) is placed in an unprimed $\langle 0, n-3, 0\rangle$ bucket, thus showing that all three sequences satisfy Property 2 of their respective sequence definitions. So all that remains is to show that Property 3 holds for each sequence.

Figures 20 and 21 show that there is exactly one way to fit the little T piece and the long L piece into an unprimed bucket, and that the result is a bucket with shape $s_P = \langle 0, 0, 0\rangle$ — and thus that the two pieces form an (s_U, s_P)-priming sequence (as well as an s_U-flattening sequence). Figure 22 shows that the sequence consisting of a short L piece, a Z piece, and a second short L piece forms a (s_U, s_P)-filler sequence. Finally, Figure 23 shows that the big T place forms an (s_U, s_P)-closing sequence. Thus, by Theorem 6.1, we know that:

Theorem 6.3. *For all $n > 5$, clearing the board in n-tris is NP-complete.*

7. Surviving with Larger Polyominoes

In the previous section, we showed that it is NP-complete to clear all rows of a given n-tris board using a given piece sequence. In this section, we extend these results to show NP-completeness of mere survival when presented with a given n-tris board and a given piece sequence. In Section 7.1, we show that, for every possible board size, there is a sequence of n-tris pieces that will result in a loss, no matter how the board is initially arranged. Then, in Section 7.2, we explain how to use this sequence, combined with the previous results about the hardness of clearing the board, to show that survival is NP-complete as well.

7.1 Unwinnable *n*-tris Sequences

In this section, we present a general unwinnable n-tris sequence for $n \geq 5$, based on the technique of Burgiel [2]. For this analysis, we ignore the shape of the piece, and examine only how the placement of each piece affects the number of filled cells added to each column.

Let s_i be the number of cells in column i that are initially filled, before playing the sequence of unwinnable pieces. Let r be the number of rows that have been cleared after playing the unwinnable sequence. For each column, let c_i be the number of filled cells in column i after playing the unwinnable sequence. Then we define $b_i = c_i + r - s_i$ to be the total number of filled cells added to column i during the unwinnable sequence. Let h be the height of the board. Then the number of filled cells in one column cannot differ by more than h from the number of filled cells in another column. That is, for all pairs of columns i, j, we must have $|c_i - c_j| < h$. We additionally must have $|s_i - s_j| < h$. Thus, we can conclude that

$$\begin{aligned}|b_i - b_j| &= |(c_i + r - s_i) - (c_j + r - s_j)| \\ &= |(c_i - c_j) + (s_j - s_i)| \\ &\leq |c_i - c_j| + |s_i - s_j| \\ &< 2h.\end{aligned}$$

We consider the case of $n = 5$ first, then explain how to generalize to $n > 5$. For $n = 5$, the unwinnable sequence consists of one piece repeated a large number of times: in particular, we use the cross piece, depicted in Figure 24. Because of the piece's symmetries, there is effectively only one orientation for the piece, which makes it easier to analyze. For each column i, let X_i be the number of crosses placed so that the center of the cross is in column i. Then we may write each b_i as a function of the values X_i:

$$b_i = X_{i-1} + 3X_i + X_{i+1}.$$

Consider the first column. Because we cannot place a cross centered on column 1 or the (non-existent) column 0, the formula is simple: $b_1 = X_2$. Now consider the difference between b_2 and b_1, which we know must be bounded by $2h$:

$$\begin{aligned}b_2 - b_1 &< 2h, \\ 3X_2 + X_3 - X_2 &< 2h, \\ 2X_2 + X_3 &< 2h.\end{aligned}$$

Because each X_i is non-negative, this means that $X_2 < 2h$. Unfortunately, this implies that $b_1 = X_2 < 2h$, so the number of filled cells inserted into the first column must be less than $2h$. Hence, the player can clear at most $b_1 + s_1 < 2h + h = 3h$ rows, meaning that they will lose as soon as they have received enough pieces to entirely fill $4h$ rows.

Now consider the case of $n > 5$. Again, we use a single piece to construct the unwinnable sequence: an asymmetric cross, shown in Figure 25. There are four orientations for the cross: northern, eastern, southern, and western. Note, however, that both northern and southern crosses result in the same number of filled cells being inserted into each column, so for simplicity we only keep track of N_i, the number of northern or southern crosses placed so that their center lies in column i. For the eastern and western crosses, we define their center to be the cell where the two bars of the cross meet. Then we define E_i and W_i to be the number of eastern (respectively, western) crosses placed so that their center lies in column i. Then we can write each b_i as a function of N_i, E_i, and W_i:

$$\begin{aligned}b_i = {} & N_{i-1} + (n-2)N_i + N_{i+1} + E_{i-1} + 3E_i \\ & + \sum_{j=1}^{n-4} E_{i+j} + W_{i+1} + 3W_i + \sum_{j=1}^{n-4} W_{i-j}.\end{aligned}$$

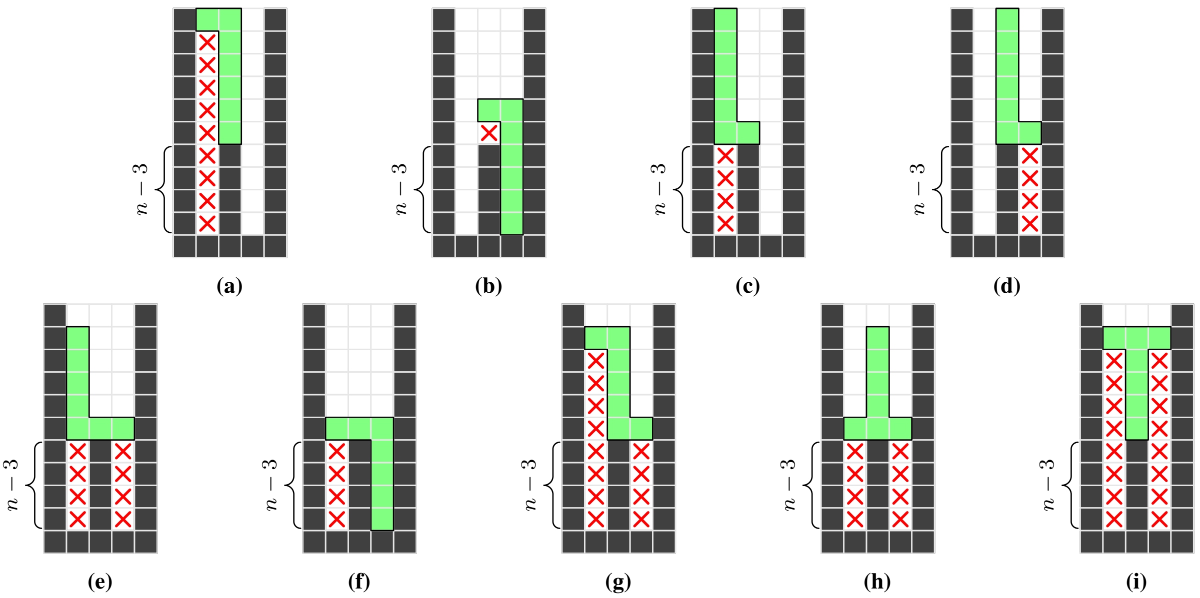


Fig. 19: Figures (a) through (d) show what happens if the long L piece (the second priming piece) is placed in an unprimed bucket. Figures (e) and (f) show what happens if the short L piece (the first and last filler piece) is used in an unprimed bucket. Figure (g) shows what happens if the Z piece (the second filler piece) is used in an unprimed bucket. Figures (h) and (i) show what happens if the big T piece (the closing piece) is used in an unprimed bucket. The holes are shaded in light red, while the choke points are labelled with a cross.

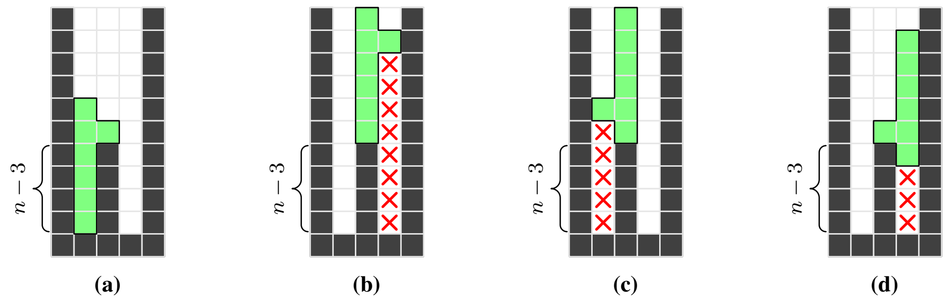


Fig. 20: Possible positions for the little T piece (the first priming piece) in an unprimed bucket.

Again, we examine the first column to compute the number of filled cells added. This time, we know that $N_i = W_i = 0$ for $i < 2$, and that $E_i = 0$ for $i < n-3$, which allows us to eliminate some terms:

$$b_1 = N_2 + E_{n-3} + W_2.$$

We then compare this to the number of cells in columns 2 and $n-3$:

$$\begin{aligned} b_2 &= (n-2)N_2 + N_3 + E_{n-3} + E_{n-2} + W_3 + 3W_2, \\ b_{n-3} &= N_{n-4} + (n-2)N_{n-3} + N_{n-2} + E_{n-4} \\ &\quad + 3E_{n-3} + \sum_{j=1}^{n-4} E_{n-3+j} + W_{n-2} + 3W_{n-3} + \sum_{j=1}^{n-4} W_{n-3-j}. \end{aligned}$$

Notably, b_2 contains the terms $(n-3)N_2 \geq 3N_2$ and $3W_2$, while b_{n-3} contains the term $3E_{n-3}$. Now consider the following equation:

$$b_2 + b_{n-3} - 2b_1 = (b_2 - b_1) + (b_{n-3} - 1) < 4h.$$

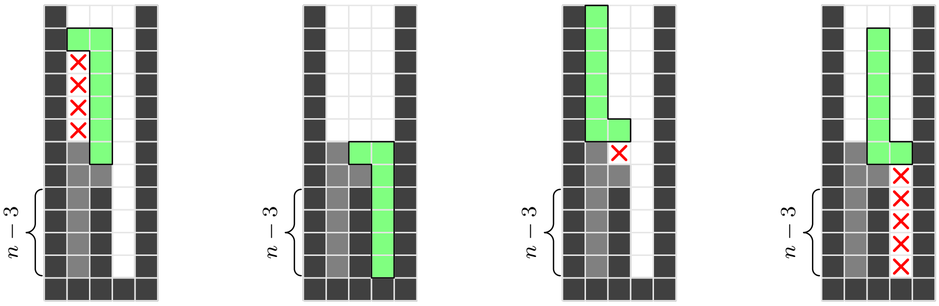
If we evaluate $b_2 + b_{n-3} - 2b_1$, we find that the term $(n-3)N_2$ will absorb the loss of $2N_2$ to become $(n-5)N_2 \geq N_2$; the term $3W_2$ will absorb the loss of $2W_2$ to become W_2; and the term $3E_{n-3}$ will absorb the loss of $2E_{n-3}$ to become E_{n-3}. Thus the result will be a sum of non-negative variables, including at least one copy of N_2, W_2, and E_{n-3}, which must be less than $4h$. Hence, we can conclude that $b_1 = N_2 + E_{n-3} + W_2 < 4h$. Thus we have shown that the player can clear at most $b_1 + s_1 < 4h + h = 5h$ rows, meaning they will lose as soon as they have received enough pieces to entirely fill $6h$ rows.

Hence, in both the case of $n = 5$ and the case of $n > 5$, we have shown the following:

Theorem 7.1. *For any $n \geq 5$, any board width w, and any board height h, the sequence consisting of $\lceil 6hw/n \rceil$ cross pieces will cause the player to lose, no matter how the board is initially configured.*

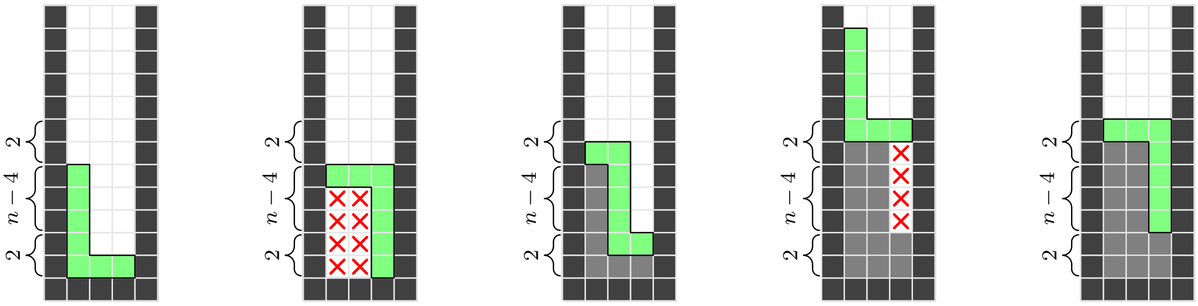
7.2 Survival Reduction

To show that survival is hard for n-tris, we reduce from 3-Partition. We begin by applying the reduction of Theorem 6.2 or Theorem 6.3, depending on whether $n = 5$. Call the resulting board B and the resulting piece sequence Q. Let w be the width of B. Let h_B be the index of the highest non-empty row of B, and let h_S be the number of empty rows above that (the number of rows



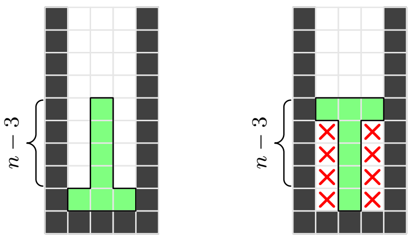
(a) **(b)** **(c)** **(d)**

Fig. 21: Possible positions for the second piece in the priming sequence.



(a) **(b)** **(c)** **(d)** **(e)**

Fig. 22: The filler sequence. Figures 22(a) and 22(b) show the two possible orientations for the first filler piece, only one of which is valid. Figure 22(c) shows how the only possible orientation for the second filler piece fits on top of the only valid position for the first filler piece. Figures 22(d) and 22(e) show the two possibly orientations for the third and final filler piece, only one of which is valid.



(a) **(b)**

Fig. 23: Possible positions for the big T piece in a primed bucket.

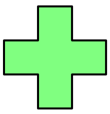
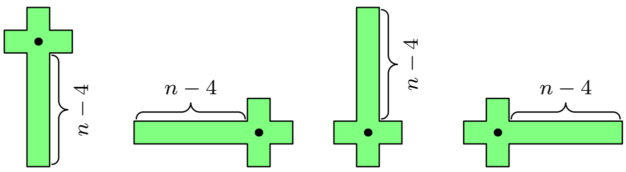


Fig. 24: The piece used to construct an unwinnable sequence for Pentris.



(a) Northern Cross. **(b)** Eastern Cross. **(c)** Southern Cross. **(d)** Western Cross.

Fig. 25: The piece used to construct an unwinnable sequence for n-tris, $n > 5$. The "center" of each is drawn with a dot. Note that this may or may not be the center of rotation of the piece — for this proof, rotation does not matter.

allocated to allow the player to move each piece to the appropriate bucket).

Let I be the $1 \times n$ piece, and let X be the cross piece (symmetric if $n = 5$, asymmetric otherwise). Let $\ell = \lceil 6(h_B + h_S)w/n \rceil$ — that is, the number of copies of X that would ensure the player loses on any board of dimensions $w \times (h_B + h_S)$ (and in particular, any way the player can reconfigure B). Then the sequence of pieces we use is QIX^l.

Just as in the original Tetris proof, we construct a new n-tris board of width w with three sections stacked vertically, as shown in Figure 26. The top part of the board consists of a copy of B. Due to the structure of the clearing reduction, the first column of B is initially filled up to height h_B. If there is a solution to the 3-Partition problem, then the player will be able to clear all h_B rows (and in particular, the entire first column) before the last I piece in the sequence. Otherwise, we know that the bottom-most cell in the first column of B will be filled when the last I piece is dropped.

Just as in the original Tetris proof in [1], we construct a new n-tris board of width w with three sections stacked vertically, as shown in Figure 26. The top part of the board consists of a copy of B. Due to the structure of the clearing reduction, the first column of B is initially filled up to height h_B. If there is a solution to the 3-Partition problem, then the player will be able to clear all h_B rows (and in particular, the entire first column) before the last I piece in the sequence. Otherwise, we know that the bottom-most cell in the first column of B will be filled when the last I piece is dropped.

The middle section of the board has height n, and is almost

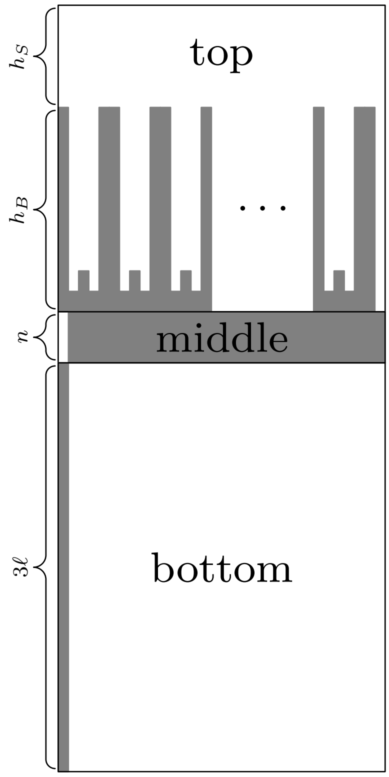


Fig. 26: The structure of the board used to show that survival is hard.

completely filled. Only the first column of the middle section is empty. If the player has used the sequence Q to clear the top part of the board (i.e. if the original 3-Partition instance is solvable), they can drop the subsequent I piece into the first column in the middle section, clearing all n rows and opening up the bottom section for their use. Otherwise, the player will only be able to use the later X pieces to try to clear the rows in the middle section. However, because of the structure of the X piece, it cannot be used to complete a row where everything but the first column is filled, so if the player cannot solve the original 3-Partition problem, they will not be able to clear any of the rows of the middle section. This means that all ℓ copies of the cross piece must be placed in the top $h_B + h_S$ rows of the board — forcing a loss.

If the player has solved the 3-Partition instance, however, then the sequence $Q \circ I$ can be used to open up the bottom part of the board, which is a space of height 3ℓ such that everything except the first column is completely clear. This leaves more than enough space to place all ℓ crosses. Thus, the player can survive if they can solve the 3-Partition instance, and they must die if they cannot solve the original 3-Partition. Hence, we have the following result:

Theorem 7.2. *For all $n \geq 5$, surviving n-tris is NP-complete.*

8. Conclusion

This paper nearly completes the taxonomy of hardness of Tetris with k-ominoes. We give a general construction for all $k > 4$, proving NP-completeness for infinitely many cases. We also show that clearing the board with dominoes with no rotation is NP-complete; clearing the board with trominoes with rotation is NP-complete; and survival or clearing the board with trominoes and no rotation is NP-complete.

We leave open a few remaining cases involving small polyominoes. Some of these cases seem deceptively simple. For example, the case of dominoes with rotation allows the pieces to fairly easily fill most spaces, but also allows the dominoes to be navigated into sections with complex internal structure. Efficient algorithms for maximizing the score with n monominoes would be also interesting. The domino and tromino cases also lack sequences of blocks that guarantee a loss for the player. This is a property happily used in other proofs to extend the hardness results from clearing to survival.

Acknowledgments We thank the anonymous referees for helpful comments.

References

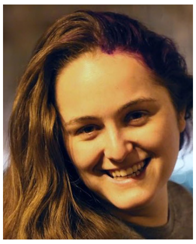
[1] Breukelaar, R., Demaine, E. D., Hohenberger, S., Hoogeboom, H. J., Kosters, W. A. and Liben-Nowell, D.: Tetris is hard, even to approximate, *Int. J. Comput. Geometry Appl.*, Vol. 14, No. 1-2, pp. 41–68 (2004).
[2] Brzustowski, J.: Can you win at Tetris?, PhD Thesis, University of British Columbia (1992).
[3] Freyer, H.: Pentris, `http://www.pentris.net`.
[4] Garey, M. R. and Johnson, D. S.: *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman & Co., New York, NY, USA (1979).
[5] Hoogeboom, H. J. and Kosters, W. A.: How to construct Tetris configurations, *International Journal of Intelligent Games & Simulation*, Vol. 3, No. 2, pp. 94–102 (2004).
[6] Kishore, S.: ntris: an idea taken a step too far, `https://blogs.oracle.com/ksplice/ntris:-an-idea-taken-a-step-too-far` (2010).
[7] Wikipedia: Tetris, `https://en.wikipedia.org/wiki/Tetris`.
[8] Yoon, A.: Tetris has 425 million paid mobile downloads, `http://www.shacknews.com/article/83932/tetris-has-425-million-paid-mobile-downloads` (2014).

Erik D. Demaine received a B.Sc. degree from Dalhousie University in 1995, and M.Math. and Ph.D. degrees from University of Waterloo in 1996 and 2001, respectively. Since 2001, he has been a professor in computer science at the Massachusetts Institute of Technology. His research interests range throughout algorithms, from data structures for improving web searches to the geometry of understanding how proteins fold to the computational difficulty of playing games. In 2003, he received a MacArthur Fellowship as a "computational geometer tackling and solving difficult problems related to folding and bending—moving readily between the theoretical and the playful, with a keen eye to revealing the former in the latter". He cowrote a book about the theory of folding, together with Joseph O'Rourke (*Geometric Folding Algorithms*, 2007), and a book about the computational complexity of games, together with Robert Hearn (*Games, Puzzles, and Computation*, 2009).

Martin L. Demaine is an artist and mathematician. He started the first private hot glass studio in Canada and has been called the father of Canadian glass. Since 2005, he has been the Angelika and Barton Weller Artist-in-Residence at the Massachusetts Institute of Technology. Both Martin and Erik work together in paper, glass, and other material. They use their exploration in sculpture to help visualize and understand unsolved problems in mathematics, and their scientific abilities to inspire new art forms. Their artistic work includes curved origami sculptures in the permanent collections of the Museum of Modern Art (MoMA) in New York, and the Renwick Gallery in the Smithsonian. Their scientific work includes over 60 published joint papers, including several about combining mathematics and art.

Sarah Eisenstat is a Ph.D. student in the Department of Electrical Engineering and Computer Science at the Massachusetts Institute of Technology, working with Professor Erik D. Demaine. Her current research interests include computational geometry and algorithms. She received her Sc.B. and Sc.M. degrees from Brown University in 2008 and 2009, respectively.

Adam Hesterberg received his A.B. degree summa cum laude from Princeton University in 2011 and is now a Ph.D. student in the Department of Mathematics at MIT, working with Professor Erik Demaine, studying computational geometry and computational complexity.



Andrea Lincoln is a Ph.D. student in the Department of Electrical Engineering and Computer Science at MIT working with Professor Virginia Vassilevska Williams. Andrea's research interests include fine grained complexity, graph algorithms and external memory algorithms. She received her S.B. and M.Eng. from MIT in 2014 and 2015 respectively. She received her Masters of Computer Science from Stanford in 2016.

Jayson Lynch is a Ph.D. student in the Department of Electrical Engineering and Computer Science at MIT working with Professor Erik Demaine. Jayson's research interests include computational geometry and origami, algorithmic reductions, and reversible computing.

Y. William Yu received his B.A. and B.S. from Indiana University in 2009. He also earned M.Res. and M.Phil. degrees from Imperial College London in 2010 and 2014 respectively. Most recently, he graduated with a Ph.D. from MIT in 2017 while studying on a Hertz Foundation Fellowship. His research interest is in algorithms for computational biology.