Tetris is Hard, Even to Approximate

Erik D. Demaine, Susan Hohenberger, and David Liben-Nowell

Laboratory for Computer Science, Massachusetts Institute of Technology
200 Technology Square, Cambridge, MA 02139, USA
{edemaine,srhohen,dln}@theory.lcs.mit.edu

Abstract. In the popular computer game of *Tetris*, the player is given a sequence of tetromino pieces and must pack them into a rectangular gameboard initially occupied by a given configuration of filled squares; any completely filled row of the gameboard is cleared and all pieces above it drop by one row. We prove that in the offline version of Tetris, it is NP-complete to maximize the number of cleared rows, maximize the number of tetrises (quadruples of rows simultaneously filled and cleared), minimize the maximum height of an occupied square, or maximize the number of pieces placed before the game ends. We furthermore show the extreme inapproximability of the first and last of these objectives to within a factor of $p^{1-\varepsilon}$, when given a sequence of p pieces, and the inapproximability of the third objective to within a factor of $2-\varepsilon$, for any $\varepsilon > 0$. Our results hold under several variations on the rules of Tetris, including different models of rotation, limitations on player agility, and restricted piecesets.

1 Introduction

Tetris [13] is a popular computer game that was invented by mathematician Alexey Pazhitnov in the mid-1980s. By 1988, just a few years after its invention, Tetris was already the best-selling game in the United States and England. Over 50 million copies have been sold worldwide. (Incidentally, Sheff [12] gives a fascinating account of the tangled legal debate over the profits, ownership, and licensing of Tetris.)

In this paper, we embark on the study of the computational complexity of playing Tetris. We consider the *offline* version of Tetris, in which the sequence of pieces that will be dropped is specified in advance. Our main result is that playing offline Tetris optimally is NP-complete, and furthermore is highly inapproximable.

The game of Tetris. Concretely, the game of Tetris is as follows. (We give precise definitions in Section 2, and discuss some variants on these definitions in Section 6.) We are given an initial *gameboard*, which is a rectangular grid with some gridsquares filled and some empty. (In typical Tetris implementations, the gameboard is 20-by-10, and "easy" levels have an initially empty gameboard, while "hard" levels have non-empty initial gameboards, usually with the gridsquares below a certain row filled independently at random.)

A sequence of *tetrominoes*—see Figure 1—is generated, typically probabilistically; the next piece appears in the middle of the top row of the gameboard. The piece falls, and as it falls the player can rotate the piece and slide it horizontally. It stops falling when it lands on a filled gridsquare, though the player has a final opportunity to slide or rotate it before it stops moving permanently. If, when the piece comes to rest, all gridsquares in an entire row h of the gameboard are filled, row h is *cleared*: all rows above h fall one row lower, and the top row of the gameboard is replaced by an entirely unfilled row. As soon as a piece is fixed in place, the next piece appears at the top of the gameboard. To assist the player, typically a one-piece *lookahead* is provided—when the ith piece begins falling, the identity of the $(i+1)$st piece is revealed.

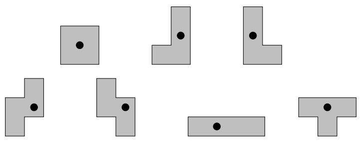

Fig. 1. *The tetrominoes* Sq *("square"),* LG *("left gun"),* RG *("right gun"),* LS *("left snake"),* RS *("right snake"),* I*, and* T*, with each piece's* center *marked.*

A player *loses* when a new piece is blocked from entirely entering the gameboard by filled gridsquares. Normally, the player can never win a Tetris game, since pieces continue to be generated until the player loses. Thus the player's objective is to maximize his or her score (which increases as pieces are placed and as rows are cleared).

Our results. In this paper, we introduce the natural full-information (offline) version of Tetris: we have a *deterministic, finite* piece sequence, and the player knows the identity and order of all pieces that will be presented. (*Games Magazine* has posed several Tetris puzzles based on the offline game [9].) We study the offline version because its hardness captures much of the difficulty of playing Tetris; intuitively, it is only easier to play Tetris with complete knowledge of the future, so the difficulty of playing the offline version suggests the difficulty of playing the online version. It also naturally generalizes the one-piece lookahead of implemented versions of Tetris.

It is natural to generalize the Tetris gameboard to m-by-n, since a relatively simple dynamic program solves the $m \cdot n = O(1)$ case in time polynomial in the number of pieces. Furthermore, in an attempt to consider the inherent difficulty of the game—and not any accidental difficulty due to the limited reaction time of the player—we initially allow the player an arbitrary number of shifts and rotations before the current piece drops by one row. (We restrict to realistic agility levels later.)

In this paper, we prove that it is NP-complete to optimize any of several natural objective functions for Tetris: (1) maximizing the number of rows cleared while playing the given piece sequence; (2) maximizing the number of pieces placed before a loss occurs; (3) maximizing the number of times a *tetris*—the simultaneous clearing of four rows—occurs; and (4) minimizing the height of the highest filled gridsquare over the course of the sequence. We also prove the extreme inapproximability of the first two (and the most natural) of these objective functions: given an initial gameboard and a sequence of p pieces, for any constant

$\varepsilon > 0$, it is NP-hard to approximate to within a factor of $p^{1-\varepsilon}$ the maximum number of pieces that can be placed without a loss, or the maximum number of rows that can be cleared. We also show that it is NP-hard to approximate the minimum height of the highest filled gridsquare to within a factor of $2 - \varepsilon$.

To prove these results, we first show that the cleared-row maximization problem is NP-hard, and then give extensions of our reduction for the remaining objectives. Our initial proof of hardness proceeds by a reduction from 3-PARTITION, in which we are given a set S of $3s$ integers and a bound T, and asked to partition S into s sets of three numbers each so that the sum of the numbers in each set is exactly T. Intuitively, we define an initial gameboard that forces pieces to be placed into s piles, and give a sequence of pieces so that all of the pieces associated with each integer must be placed into the same pile. The player can clear all rows of the gameboard if and only if all s of these piles have the same height. A key difficulty in our reduction is that there are only a constant number of piece types, so any interesting component of a desired NP-hard problem instance must be encoded by a sequence of multiple pieces. The bulk of our proof of correctness is devoted to showing that, despite the decoupled nature of a sequence of Tetris pieces, the only way to possibly clear the entire gameboard is to place in a single pile all pieces associated with a particular integer.

Our reduction is robust to a wide variety of modifications to the rules of the game. In particular, our results continue to hold in the following settings: (1) with restricted player agility—allowing only two rotation/translation moves before each piece drops in height; (2) under a wide variety of different rotation models—including the somewhat non-intuitive model that we have observed in real Tetris implementations; (3) without any losses—i.e., with an infinitely tall gameboard; and (4) when the pieceset is restricted to $\{\mathsf{LG}, \mathsf{LS}, \mathsf{I}, \mathsf{Sq}\}$ or $\{\mathsf{RG}, \mathsf{RS}, \mathsf{I}, \mathsf{Sq}\}$, plus at least one other piece.

Related work: Tetris. This paper is, to the best of our knowledge, the first consideration of the complexity of playing Tetris. Kostreva and Hartman [10] consider Tetris from a control-theoretic perspective, using dynamic programming to choose the "optimal" move, using a heuristic measure of configuration quality. Other previous work has concentrated on the possibility of a *perpetual loss-avoiding strategy* in the online, infinite version of the game. In other words, under what circumstances can the player be forced to lose, and how quickly? Brzustowski [2] has characterized all one-piece (and some two-piece) piecesets for which there are perpetual loss-avoiding strategies. He has also shown that, if the machine can adversarially choose the next piece (following the lookahead piece) in reaction to the player's moves, then the machine can force an eventual loss using any pieceset containing $\{\mathsf{LS}, \mathsf{RS}\}$. Burgiel [3] has strengthened this result, showing that an alternating sequence of LS's and RS's will eventually cause a loss in any gameboard of width $2n$ for odd n, regardless of the player's strategy. This implies that, if pieces are chosen independently at random with non-zero probability mass on both LS and RS, there is a forced eventual loss with probability one.

Recently, Breukelaar, Hoogeboom, and Kosters [1] have given a significant simplification of our reduction and proof of the NP-hardness of maximizing the number of rows cleared in a Tetris game. By using a more restrictive construction to limit piece placement, they are able to give a much shorter proof that all pieces associated with a particular integer must be placed in the same pile. (They have many fewer cases to consider.) The extensions to our reduction that we present in Sections 4, 5, and 6 can also be applied to their reduction to achieve the same results regarding different rules/objectives and inapproximability.

Related work: other games and puzzles. A number of other popular one-player computer games have recently been proven to be NP-hard, most notably the game of Minesweeper [8]—or, more precisely, the Minesweeper "consistency" problem. See the survey of the first author [4] for a summary of other games and puzzles that have been studied from the perspective of computational complexity. These results form the emerging area of *algorithmic combinatorial game theory*, in which many new results have been established in the past few years, e.g., Zwick's positive results on optimal strategies for the two-player block-stacking game *Jenga* [14].

2 Rules of Tetris

Here we rigorously define the game of Tetris, formalizing the intuition of the previous section. For concreteness, we have chosen to give very specific rules, but in fact the remainder of this paper is robust to a variety of modifications to these rules; in Section 6, we will discuss some variations on these rules for which our results still apply.

The *gameboard* is a grid of m rows and n columns, indexed from bottom-to-top and left-to-right. The $\langle i, j \rangle$th *gridsquare* is either *unfilled* (*open*, *unoccupied*) or *filled* (*occupied*). In a legal gameboard, no row is completely filled, and there are no completely empty rows that lie below any filled gridsquare. When determining the legality of certain moves, we consider all gridsquares outside the gameboard as always-occupied sentinels.

The seven Tetris pieces are exactly those connected rectilinear polygons that can be created by assembling four 1-by-1 gridsquares. The *center* of each piece is shown in Figure 1. A *piece state* $P = \langle t, o, \langle i, j \rangle, f \rangle$ consists of: (1) a *piece type* $t \in \{\mathsf{Sq}, \mathsf{LG}, \mathsf{RG}, \mathsf{LS}, \mathsf{RS}, \mathsf{I}, \mathsf{T}\}$; (2) an *orientation* $o \in \{0°, 90°, 180°, 270°\}$, the number of degrees clockwise from the piece's *base orientation* (shown in Figure 1); (3) a *position* $\langle i, j \rangle \in \{1, \ldots, m\} \times \{1, \ldots n\}$ of the piece's center on the gameboard; and (4) the value $f \in \{\mathit{fixed}, \mathit{unfixed}\}$, indicating whether the piece can continue to move. (The position of a Sq is the location of the upper-left gridsquare of the Sq, since its center falls on the boundary of four gridsquares rather than in the interior of one.) In an *initial piece state*, the piece is in its base orientation, and the initial position places the highest gridsquares of the piece into row m, and the center into column $\lfloor n/2 \rfloor$, and the piece is unfixed.

For now, rotations will follow the *instantaneous rotation model*. (We discuss other rotation models in Section 6.) For a piece state $P = \langle t, o, \langle i, j \rangle, \mathit{unfixed} \rangle$, a gameboard B, and a rotation angle $\theta = \pm 90°$, the rotated piece state $R(P, \theta, B)$ is $\langle t, (o + \theta) \bmod 360°, \langle i, j \rangle, \mathit{unfixed} \rangle$ as long as all the gridsquares occupied by the rotated piece are unoccupied in B; if some of these gridsquares are full in B then $R(P, \theta, B) = P$ and the rotation is *illegal*.

Playing the game. No moves are legal for a piece $P = \langle t, o, \langle i, j \rangle, \mathit{fixed} \rangle$. The following moves are legal for a piece $P = \langle t, o, \langle i, j \rangle, \mathit{unfixed} \rangle$, with current gameboard B: (1) a *rotation*, resulting in the piece state $R(P, \pm 90°, B)$; (2) a *translation*, resulting in the piece state $\langle t, o, \langle i, j \pm 1 \rangle, \mathit{unfixed} \rangle$, if the gridsquares adjacent to P are open in B; (3) a *drop*, resulting in the piece state $\langle t, o, \langle i - 1, j \rangle, \mathit{unfixed} \rangle$, if all the gridsquares beneath P are open in B; and (4) a *fix*, resulting in $\langle t, o, \langle i, j \rangle, \mathit{fixed} \rangle$, if at least one gridsquare below P is occupied in B. A *trajectory* σ of a piece P is a sequence of legal moves starting from an initial state and ending with a fix move. The result of this trajectory on gameboard B is a new gameboard B', as follows:

1. The new gameboard B' is initially B with the gridsquares of P filled.
2. If the piece is fixed so that, for some row r, every gridsquare in row r of B' is full, then row r is *cleared*. For each $r' \geq r$, replace row r' of B' by row $r' + 1$ of B'. Row m of B' is an empty row. Multiple rows may be cleared by the fixing of a single piece.
3. If the next piece's initial state is blocked in B', the game ends and the player *loses*.

For a *game* $\langle B_0, P_1, \ldots, P_p \rangle$, a *trajectory sequence* Σ is a sequence $B_0, \sigma_1, B_1, \ldots, \sigma_p, B_p$ so that, for each i, the trajectory σ_i for piece P_i on gameboard B_{i-1} results in gameboard B_i. However, if there is a losing move σ_q for some $q \leq p$ then the sequence Σ terminates at B_q instead of B_p.

The Tetris problem. For concreteness, we will focus our attention on the following Tetris problem: given a Tetris game $\mathcal{G} = \langle B, P_1, P_2, \ldots, P_p \rangle$, does there exist a trajectory sequence Σ that clears the entire gameboard of $\mathcal{G}$? (We will consider other Tetris objectives in Section 4.) Membership of Tetris in NP follows straightforwardly.

3 NP-completeness of Tetris

We define a mapping from instances of 3-Partition [7, p. 224] to instances of Tetris. Recall the 3-Partition problem:

Given: A sequence $a_1, \ldots, a_{3s}$ of non-negative integers and a non-negative integer T, so that $T/4 < a_i < T/2$ for all $1 \leq i \leq 3s$ and so that $\sum_{i=1}^{3s} a_i = sT$.
Output: Can $\{1, \ldots, 3s\}$ be partitioned into s disjoint subsets $A_1, \ldots, A_s$ so that, for all $1 \leq j \leq s$, we have $\sum_{i \in A_j} a_i = T$?

We choose to reduce from 3-Partition because it is NP-hard to solve this problem even if the inputs a_i and T are provided in unary:

Theorem 1 (Garey and Johnson [6]). 3-PARTITION *is* NP-*complete in the strong sense.* □

Given an arbitrary instance $\mathcal{P} = \langle a_1, \ldots, a_{3s}, T \rangle$ of 3-PARTITION, we will produce a Tetris game $\mathcal{G}(\mathcal{P})$ whose gameboard can be completely cleared precisely if $\mathcal{P}$ is a "yes" instance. (For brevity, we omit some details; see [5].)

The initial gameboard is shown in Figure 2. The topmost $3s + O(1)$ rows form an empty staging area for rotations and translations. Below, there are s *buckets*, each six columns wide, corresponding to the sets $A_1, \ldots, A_s$ for the instance of 3-PARTITION. Each bucket has unfilled *notches* in its fourth and fifth columns in every sixth row, beginning in the fifth row. The first four rows of the first and second columns in each bucket are initially filled, and the sixth column of each bucket is entirely filled. The last three columns of the gameboard form a *lock*, blocking access to the last column, which is unfilled in all rows but the second-highest. Until a piece is placed into the lock to clear the top two rows, no lower rows can be cleared.

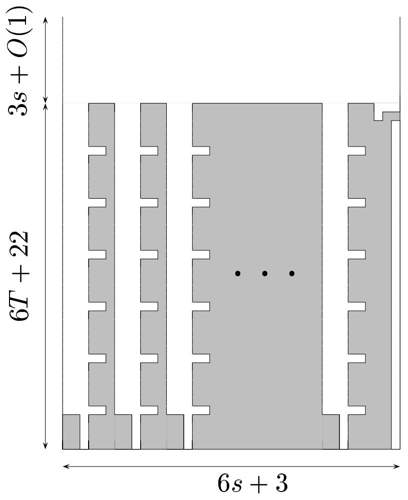

Fig. 2. *The initial gameboard for a Tetris game mapped from an instance of* 3-PARTITION.

The piece sequence consists of the following sequence of pieces for each $a_1, \ldots, a_{3s}$: one *initiator* $\langle \mathsf{I}, \mathsf{LG}, \mathsf{Sq} \rangle$, then a_i repetitions of the *filler* $\langle \mathsf{LG}, \mathsf{LS}, \mathsf{LG}, \mathsf{LG}, \mathsf{Sq} \rangle$, and then one *terminator* $\langle \mathsf{Sq}, \mathsf{Sq} \rangle$. After the pieces associated with $a_1, \ldots, a_{3s}$, we have the following additional pieces: s successive I's, one RG, and $3T/2 + 5$ successive I's. (Without loss of generality, we can assume T is even by multiplying all input numbers by two.)

The TETRIS instance $\mathcal{G}(\mathcal{P})$ has size polynomial in the size of the 3-PARTITION instance $\mathcal{P}$, since $a_1, \ldots, a_{3s}$ and T are represented in unary, and can be constructed in polynomial time.

Lemma 2 (Completeness). *For any "yes" instance $\mathcal{P}$ of* 3-PARTITION*, there is a trajectory sequence Σ that clears the entire gameboard of $\mathcal{G}(\mathcal{P})$ without triggering a loss.*

Proof. In Figure 3, we show how to place all of the pieces associated with the number a_i in a bucket. Since $\mathcal{P}$ is a "yes" instance, there is a partitioning of $\{1, \ldots, 3s\}$ into sets $A_1, \ldots, A_s$ so that $\sum_{i \in A_j} a_i = T$. Place all pieces associated with each $i \in A_j$ into the jth bucket of the gameboard. This yields a configuration in which only the last four rows of the third column of each bucket are unfilled. Next we place one of the s successive I's into each bucket, and the RG into the lock; the first two rows are then cleared. Finally, we place the

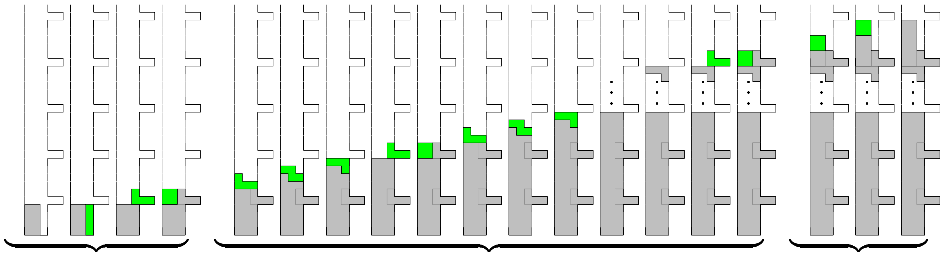

Fig. 3. *A valid sequence of moves within a bucket.*

$3T/2 + 5$ successive I's into the last column. Each of the I's clears four rows; in total, this clears the entire gameboard. □

The proof of soundness is somewhat more involved; here we give a high-level summary and some suggestive details only. Call a trajectory sequence *valid* if it completely clears the gameboard of $\mathcal{G}(\mathcal{P})$, and call a bucket *unfillable* if it is impossible to fill all of the empty gridsquares in it using arbitrarily many pieces from the set $\{\mathsf{LG}, \mathsf{LS}, \mathsf{Sq}, \mathsf{I}\}$. Also, we say that a configuration with all buckets as in Figure 4 is *unprepped.*

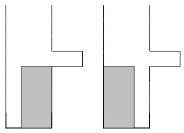

Fig. 4. *Unprepped buckets.*

Proposition 3. *In any valid trajectory sequence:*

1. *no gridsquare above row* $6T + 22$ *is ever filled;*
2. *all gridsquares of all pieces preceding the* RG *must all be placed into buckets, filling all empty bucket gridsquares;*
3. *no rows are cleared before the* RG *in the sequence;*
4. *all gridsquares of all pieces starting with (and including) the* RG *must be placed into the lock columns, filling all empty lock gridsquares;*
5. *no configuration with an unfillable bucket arises.*
6. *in an unprepped configuration, all pieces in the sequence* $\mathsf{I}, \mathsf{LG}, \mathsf{Sq}, r \times \langle \mathsf{LG}, \mathsf{LS}, \mathsf{LG}, \mathsf{LG}, \mathsf{Sq} \rangle, \mathsf{Sq}, \mathsf{Sq}$*, for any* $r \geq 1$*, must be placed into a single bucket, yielding an unprepped configuration.*

Proof. For (1), there are only enough pieces to fill and clear the gameboard if every filled gridsquare (from the initial gameboard or from pieces in the sequence) is placed into the lowest $6T + 22$ rows. For (2) and (3), placing any piece other than RG as the first piece to enter the lock columns violates (1), and no row can be cleared until some piece enters the lock. We have (4) from (1,2) and the fact that there are exactly as many gridsquares following the RG as there are empty gridsquares in the lock columns. Finally, (5) follows immediately from (2,3).

The (tedious) details for (6) can be found in [5]; here we give a high-level overview. Call a trajectory sequence *deviating* if it does not place all the pieces into the same bucket to yield an unprepped configuration. We first catalogue ten different classes of buckets that we show to be unfillable. (For example, a bucket with a disconnected region of unfilled gridsquares is unfillable.) We then exhaustively consider the tree of all possible placements of the pieces in the given sequence into the gameboard, and show that an unfillable bucket is produced by every deviating trajectory sequence. The overwhelming majority of deviating trajectory sequences create an unfillable bucket with their first deviating move, while some do not create an unfillable bucket until up to five pieces later. □

Lemma 4 (Soundness). *If there is a valid trajectory sequence for $\mathcal{G}(\mathcal{P})$, then $\mathcal{P}$ is a "yes" instance of* 3-PARTITION.

Proof. If there is a valid strategy for $\mathcal{G}(\mathcal{P})$, then by Proposition 3.2, there is a way of placing all pieces preceding the RG to exactly fill the buckets. By Proposition 3.6, we must place all of the pieces associated with a_i into the same bucket; by Proposition 3.1, we must have exactly the same number of filled gridsquares in each bucket. Define $A_j := \{i : \text{all the pieces associated with } a_i \text{ are placed into bucket } j\}$. The total number of gridsquares placed into each bucket is the same; because every $a_i \in (T/4, T/2)$, we have that each $|A_j| = 3$. Thus the A_j's form a legal 3-partition, and $\mathcal{P}$ is a "yes" instance. □

Theorem 5. *Maximizing the number of rows cleared in a Tetris game is* NP-*complete.* □

4 NP-hardness for Other Objectives

In this section, we sketch reductions extending that of Section 3 to establish the hardness of optimizing several other natural Tetris objectives. It is easy to confirm that TETRIS remains in NP for all objectives considered below.

Theorem 6. *Maximizing the number of tetrises (the number of times that four rows are cleared simultaneously) in a Tetris game is* NP-*complete.*

Proof. Our gameboard, shown in Figure 5, is augmented with four new bottom rows that are full in all but the sixth column. We append a single I to our previous piece sequence. For a "yes" instance of 3-PARTITION, $(6T + 20)/4 + 1$ tetrises are achievable. For a "no" instance, we cannot clear the top $6T + 22$ rows using the original pieces and thus we clear at most $6T + 22$ total rows. This implies that there were at most $(6T + 20)/4 < (6T + 20)/4 + 1$ tetrises. (Recall T is even.) Therefore we can achieve $(6T+24)/4$ tetrises exactly when the top $6T+22$ rows exactly when the 3-PARTITION instance is a "yes" instance. □

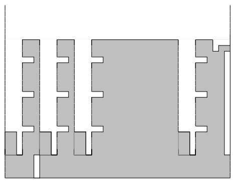

Fig. 5. *Gameboard for the hardness of maximizing tetrises.*

A different type of objective—considered by Brzustowski [2] and Burgiel [3], for example—is that of *survival.* How many pieces can be placed before a loss must occur? Our original reduction yields some initial intuition on the hardness of maximizing lifetime. In the "yes" case of 3-PARTITION, there is a trajectory sequence that fills no gridsquares above the $(6T + 22)$nd row, while in the "no" case we must fill some gridsquare in the $(6T + 23)$rd row:

Theorem 7. *Minimizing the maximum height of a filled gridsquare in a Tetris game is* NP*-complete.* □

However, this does not imply the hardness of maximizing the number of pieces that can be placed without losing, because Theorem 7 only applies for certain heights—and, in particular, does not apply for height m, because the trajectory sequence from Lemma 2 requires space above the $(6T+22)$nd row for rotations and translations. To show the hardness of maximizing survival time, we need to do some more work.

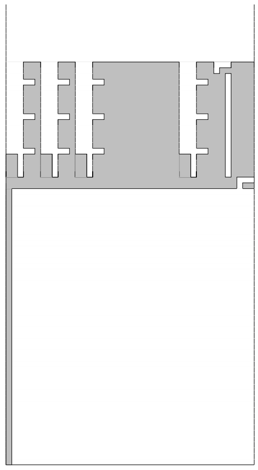

Fig. 6. *Gameboard for hardness of maximizing survival time.*

Theorem 8. *Maximizing the number of pieces placed without losing is* NP*-complete.*

Proof. We augment our previous reduction as shown in Figure 6. We have created a large *reservoir* of r rows filled only in the first column, and a second *lock* in four new columns, which prevents access to the reservoir until all the top rows are cleared. We append to our piece sequence a single RG (to open the lock) and enough Sq's to completely fill the reservoir. Choose r so that the unfilled area R of the reservoir is more than twice the total area A of the remainder of the gameboard. Observe that the gameboard has odd width, so the unfilled block of the reservoir has even width.

In the "yes" case of 3-PARTITION, we can clear the top part of the gameboard as before, then open the lock using the RG, and then completely fill and clear the reservoir using the Sq's.

In the "no" case, we cannot entirely clear the top part, and thus cannot unlock the reservoir with the RG. No number of the Sq's can ever subsequently clear the lower lock row. We claim that a loss will occur before all of the Sq's are placed. There are an odd number of columns, so only rows that initially contain an odd number of filled gridsquares can be cleared by the Sq's. Thus each row in the top of the gameboard can be cleared at most once; this uses fewer than half of the Sq's. The remainder of the Sq's cover $R/2 > A$ area, and never clear rows. Thus they must cause a loss. □

5 Hardness of Approximation

Theorem 9. *Given a game consisting of p pieces, approximating the maximum number of rows that can be cleared to within a factor of $p^{1-\varepsilon}$ for any constant $\varepsilon > 0$ is* NP*-hard.*

Proof. Our construction is as in Figure 6, with $r > a^{2/\varepsilon}$ rows in the reservoir, where there are a total rows at or above the second lock. As before, we append to the original piece sequence one RG followed by exactly enough Sq's to completely fill the reservoir. As in Theorem 8, in the "yes" case of 3-PARTITION, we can clear the entire gameboard (including the r rows of the reservoir), while in the "no" case case we can clear at most a rows. Thus it is NP-hard to distinguish the case in which at least r rows can be cleared from the case in which at most $a < r^{\varepsilon/2}$ rows can be cleared.

Note that the number of columns c in our gameboard is fixed and independent of r, and that the number of pieces in the sequence is constrained by $r < p < (r+a)c$. We also require that r be large enough that $p < (r+a)c < r^{2/(2-\varepsilon)}$. (Note that r, and thus our game, is still polynomial in the size of the 3-PARTITION instance.) Thus in the "yes" case we clear at least $r > p^{1-\varepsilon/2}$ rows, and in the "no" case we clear at most $a < r^{\varepsilon/2} < p^{\varepsilon/2}$. Thus it is NP-hard to approximate the number of cleared rows to within a factor of $(p^{1-\varepsilon/2})/(p^{\varepsilon/2}) = p^{1-\varepsilon}$. □

Using the construction in Figure 6 (with appropriate choice of r), similar arguments yield the following inapproximability results [5]:

Theorem 10. *Given a game consisting of p pieces, approximating the maximum number of pieces that can be placed without a loss to within a factor of $p^{1-\varepsilon}$ for any constant $\varepsilon > 0$ is* NP*-hard.* □

Theorem 11. *Given a game consisting of p pieces, approximating the minimum height of the highest filled gridsquare to within a factor of $2 - \varepsilon$ for any constant $\varepsilon > 0$ is* NP*-hard.* □

6 Varying the Rules of Tetris

Because the completeness of our reduction does not depend on the full set of allowable moves in Tetris—nor soundness on all limitations—our results continue to hold in some modified settings.

In real implementations of Tetris, there is a fixed amount of time (varying with the difficulty level) in which to make manipulations at height h. We consider players with limited dexterity, who can only make a small number of translations and rotations before the piece drops another row.

We defined a loss as the fixing of a piece so that it does not fit entirely within the gameboard. Other models also make sense: e.g., we might define a loss as occurring only *after* rows have been cleared—that is, a piece *can* be fixed so that it extends into the would-be $(m+1)$st row of the m-row gameboard, so long as this is not the case once all filled rows are cleared.

Finally, we consider a broad class of *reasonable* models for piece rotation. Three models of particular interest are: (1) the instantaneous model of Section 2; (2) the *continuous* (or *Euclidean*) rotation model—the natural model if one pictures pieces physically rotating in space—which extends the instantaneous model by requiring that all gridsquares that a piece *passes through* during its rotation be unoccupied; and (3) the *Tetris rotation model*, illustrated in Figure 7, which we have observed in a number of actual Tetris implementations. In this model, the position of a piece in a particular orientation is determined as follows: within the pictured k-by-k square (fixed independently of orientation), the piece is positioned so that the smallest rectangle bounding it is centered in the square, shifted upwards and leftwards as necessary to align it with the grid. (Incidentally, it took us some time to realize that the "real" rotation in Tetris did not follow the instantaneous model, which is intuitively the most natural one.)

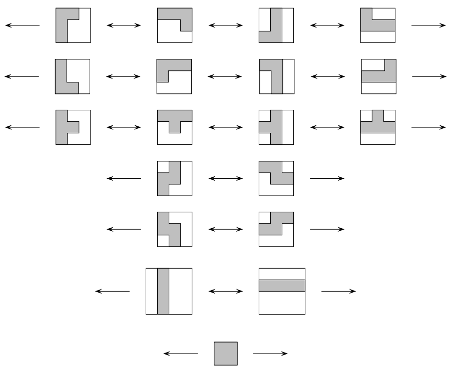

Fig. 7. *The Tetris model of rotation. Each piece can be rotated clockwise (respectively, counterclockwise) to yield the configuration on its right (respectively, left).*

Theorem 12. *It remains* NP*-hard to optimize (or approximate) the maximum height of a filled gridsquare or the number of rows cleared, tetrises attained, or pieces placed without a loss when any of the following hold:*

1. *the player is restricted to two rotation/translation moves before each piece drops in height.*
2. *pieces are restricted to* {LG, LS, I, Sq} *or* {RG, RS, I, Sq} *plus one other piece.*
3. *losses are not triggered until* after *filled rows are cleared, or no losses occur at all. (In the latter case, the objective of Theorem 8 is irrelevant.)*
4. *rotations follow any reasonable rotation model.*

Proof. For (1), note that completeness (Lemma 2) requires only two translations (to slide a LG into a notch) before the piece falls; soundness (Lemma 4) cannot be falsified by restricting legal moves. For (2), observe that our reduction uses only the pieces {LG, LS, I, Sq, RG}. For the other piecesets, we can take the mirror image of our reduction and/or replace the RG and the lock—our sole requirement on the lock is that that it be opened by a piece that does not appear elsewhere in the sequence. Claim (3) follows since we do not depend on the definition of losses in our proof: the completeness trajectory sequence does not approach the top of the gameboard, and the proof of soundness relies only on unfillability (and not on losses). Finally, for (4), our proof of Lemma 4 assumes an arbitrary reasonable rotation model [5]. Completeness follows even with the reasonability conditions, since the only rotations required for Lemma 2 occur in the upper staging area of the gameboard. □

7 Future Directions

An essential part of our reduction is a complicated initial gameboard; it is a major open question whether Tetris can be played efficiently with an empty initial configuration.

Our results are largely robust to variations on the rules (see Section 6), but our completeness result relies on the translation of pieces as they fall. At more difficult levels of the game, it may be very hard to make two translations before the piece drops another row in height. Suppose each piece can be translated and rotated as many times as the player pleases, and then falls into place [2]; no manipulations are allowed after the first downward step. Is the game still hard?

It is also interesting to consider Tetris with gameboards of restricted size. What is the complexity of Tetris for an m-by-n gameboard with $m = O(1)$ or $n = O(1)$? Is Tetris fixed-parameter tractable with respect to either m or n? (We have polynomial-time algorithms for the special cases in which $m \cdot n$ is logarithmic in the number of pieces in the sequence, or for the case of $n = 2$.)

We have reduced the pieceset down to five of the seven pieces. For what piecesets is Tetris polynomial-time solvable? (E.g., the pieceset $\{\mathsf{I}\}$ seems polynomially solvable, though non-trivial because of the initial partially filled gameboard.)

Finally, in this paper we have concentrated our efforts on the offline, adversarial version of Tetris. In a real Tetris game, the initial gameboard and piece sequence are generated probabilistically, and the pieces are presented in an online fashion. What can be said about the difficulty of playing online Tetris if pieces are generated independently at random according to the uniform distribution, and the initial gameboard is randomly generated? Some possible directions for this type of question have been considered by Papadimitriou [11].

Acknowledgments. We would like to thank Amos Fiat and Ming-wei Wang for helpful initial discussions, Chris Peikert for comments on an earlier draft, and Josh Tauber for pointing out the puzzles in *Games Magazine* [9]. This work was partially supported by NDSEG and NSF Graduate Research Fellowships.

References

1. R. Breukelaar, H. J. Hoogeboom, and W. A. Kosters. Tetris is hard, made easy. Technical report, Leiden Institute of Advanced Computer Science, 2003.
2. J. Brzustowski. Can you win at Tetris? Master's thesis, U. British Columbia, 1992.
3. H. Burgiel. How to lose at Tetris. *Mathematical Gazette*, July 1997.
4. E. D. Demaine. Playing games with algorithms: Algorithmic combinatorial game theory. In *Proc. MFCS*, pages 18–32, August 2001. `cs.CC/0106019`.
5. E. D. Demaine, S. Hohenberger, and D. Liben-Nowell. Tetris is hard, even to approximate. Technical Report MIT-LCS-TR-865, 2002. `cc.CC/0210020`.
6. M. R. Garey and D. S. Johnson. Complexity results for multiprocessor scheduling under resource constraints. *SIAM J. Comput.*, 4:397–411, 1975.
7. M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman and Company, New York, 1979.
8. R. Kaye. Minesweeper is NP-Complete. *Math. Intelligencer*, 22(2):9–15, 2000.
9. S. Kim. Tetris unplugged. *Games Magazine*, pages 66–67, July 2002.
10. M. M. Kostreva and R. Hartman. Multiple objective solution for Tetris. Technical Report 670, Department of Mathematical Sciences, Clemson U., May 1999.
11. C. Papadimitriou. Games against nature. *J. Comp. Sys. Sci.*, 31:288–301, 1985.
12. D. Sheff. *Game Over: Nintendo's Battle to Dominate an Industry.* Hodder and Stoughton, London, 1993.
13. Tetris, Inc. `http://www.tetris.com`.
14. U. Zwick. Jenga. In *Proc. SODA*, pages 243–246, 2002.