Tetris is Hard, Made Easy

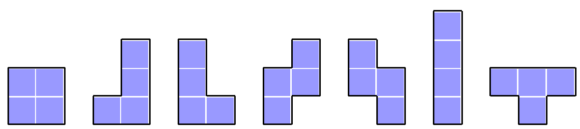
Ron Breukelaar, Hendrik Jan Hoogeboom, and Walter A. Kosters

Leiden Institute of Advanced Computer Science
Universiteit Leiden
P.O. Box 9512, 2300 RA Leiden, The Netherlands
{rbreukel,hoogeboo,kosters}@liacs.nl

Abstract. In their paper "Tetris is Hard, Even to Approximate" [2] Demaine, Hohenberger and Liben-Nowell show that optimally playing the "offline" version of Tetris, where the initial board and piece sequence are known, is NP-hard. This is done by reducing the so-called 3-PARTITION problem to this "offline" TETRIS problem. In this document a much simpler way to accomplish the same is suggested.

1 Introduction

In this paper we are concerned with the complexity of the game of Tetris. Recently, interest to the algorithmic theory of games has grown, see, e.g., [1]. The Tetris game is played with pieces as shown below:



Starting from a given partially filled board (like the one in Section 2.3) the problem is to exactly fill the remaining empty spaces with pieces from a given sequence, where these pieces "fall down" from above, and where rows containing no more empty spaces are removed.

The paper "Tetris is Hard, Even to Approximate" by Demaine, Hohenberger and Liben-Nowell [2] drew some attention in the media. Their main result is that optimally playing the "offline" version of Tetris, where the initial board and piece sequence are known, is NP-hard. This is done by reducing the known problem 3-PARTITION to this "offline" problem TETRIS. In this document a much simpler way to accomplish the same is suggested. By defining an easier reduction, a smaller board and a smaller sequence of Tetris pieces it will be easier to prove that the reduction holds for several Tetris rule-sets. By taking 'buckets' of two columns in width it is virtually impossible to rotate pieces and the number of possible piece placements is reduced enormously. It is therefore not necessary anymore to consider special rotation models.

For a precise definition of the game and the TETRIS problem the reader is referred to [2].

The authors would like to thank Siegfried Nijssen for his helpful input and comments.

2 Reduction

To prove that Tetris is NP-hard we reduce 3-PARTITION to TETRIS (as is done in [2]). By proving that every instance of 3-PARTITION can be suitably represented by a TETRIS instance constructed in polynomial time in the size of the instance, we show that TETRIS is of the same “hardness” as 3-PARTITION.

2.1 The Tetris problem

In this paper we examine the “offline” TETRIS problem as is described precisely in Section 2 from [2]. In short this means:

Given An initial game board and a finite sequence of Tetris pieces.

Question Can the Tetris pieces, supplied in the order given, be translated and rotated such that the game board will be cleared with these pieces, the last piece of the sequence filling the final gap?

Clearly, this problem is in the class NP. Note that if the game board can be cleared with the supplied Tetris pieces, it is easy to derive the number of cleared rows. Therefore this definition does not differ from “max-cleared-rows” as used in [2].

2.2 The 3-Partition problem

The problem 3-PARTITION can be defined as follows:

Given A sequence A of positive integer values $a_1, \ldots, a_{3s}$ and a positive integer value T such that $T/4 < a_i < T/2$ for all $1 \leq i \leq 3s$, and such that $\sum_{i=1}^{3s} a_i = s\,T$.

Question Can A be divided into s disjoint subsets (or rather subsequences) $B_1, \ldots, B_s$ such that: $\sum_{a_i \in B_j} a_i = T$ for all $1 \leq j \leq s$? (Call (A, T) a “yes” instance if this is the case and a “no” instance otherwise.)

Note that because $T/4 < a_i < T/2$ for all $1 \leq i \leq 3s$, in a “yes” instance $|B_j| = 3$ for all $1 \leq j \leq s$.

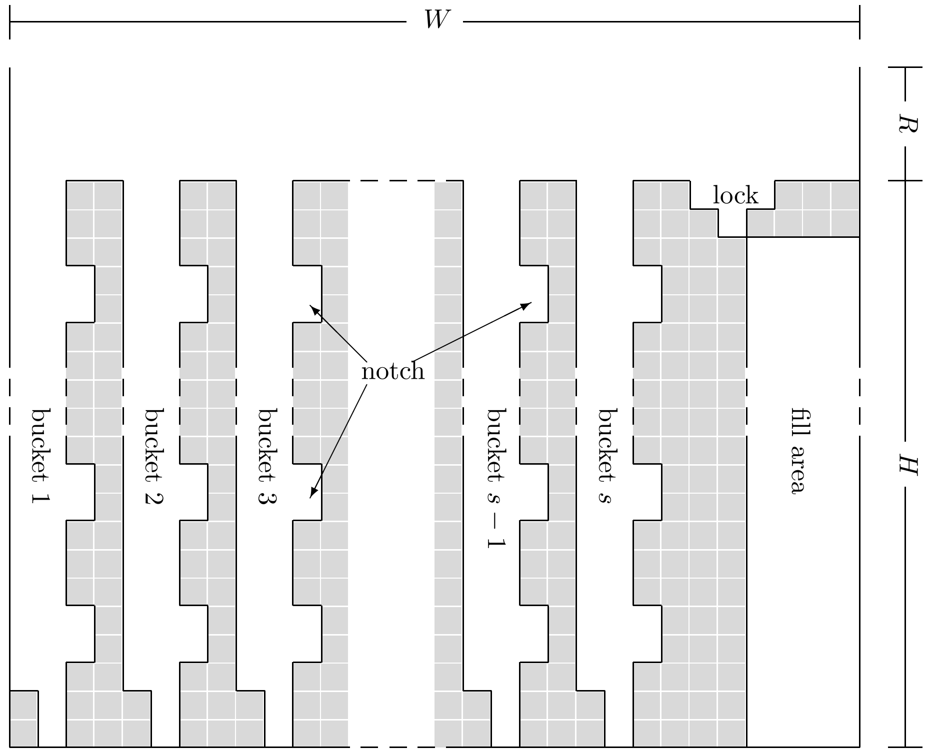
We use the following result:

Theorem 1. *(Garey and Johnson [3, p. 99])* 3-PARTITION *is NP-complete in the strong sense.*

To reduce an instance (A, T) of this problem to an instance of TETRIS we will define an initial game board and a sequence of Tetris pieces such that (A, T) is a “yes” instance if and only if the corresponding Tetris game board can be cleared with the given pieces.

2.3 The initial Tetris game board

The initial Tetris game board used in our reduction looks like this:



Its dimensions are as follows:

- R is the space needed to rotate and translate the pieces. We consider R to be big enough to rotate and translate Tetris pieces above the 'buckets' and therefore R is of no consequence to the reduction.
- W is the width of the game board and is equal to $4s + 6$.
- H is the height of the bottom part of the game board that needs to be cleared and is equal to $5T + 18$.

Note that the board is constructable in polynomial time (measured in the input size), since the variables in the problem definition may be given in unary due to the strong sense of NP-completeness (Theorem 1). On its constructibility, consult [4].

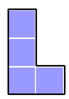
In order to explain the reduction and following [2], we call the empty columns 'buckets', the big rectangular space on the right 'fill area', the little T-shape on the right-top of the board the 'lock'; every 'bucket' has $T + 3$ 'notches' in its right side.

Every 'bucket' represents a subset in 3-Partition. There are s 'buckets' just like there are s subsets in 3-Partition. The 'fill area' and the 'lock' ensure that none of the H lines of the board will be cleared before the 'lock' is cleared and the 'notches' ensure that there is only one way to fill the 'buckets'. The shape of the 'fill area' is not very important, but its height must be $H - 2$ and it should be "fillable" with every reduction.

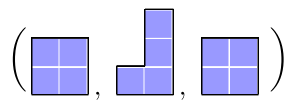
2.4 The sequence of Tetris pieces

From an instance of 3-Partition a sequence of Tetris pieces is constructed in the following way:

1. First for every $a_i \in A$ the sequence (in this order):

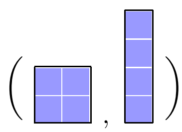
   

   'begin' followed by a_i times

   'middle',

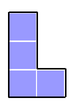
   followed by

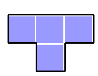
   'end'

2. Then to fill the top of all the s buckets the 'subset fillers':

   s times

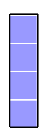
   

3. Then the T-shape to unlock the 'lock':

4. And to clear the whole board by filling the 'fill area':

   $5T + 16$ times

3 Proof

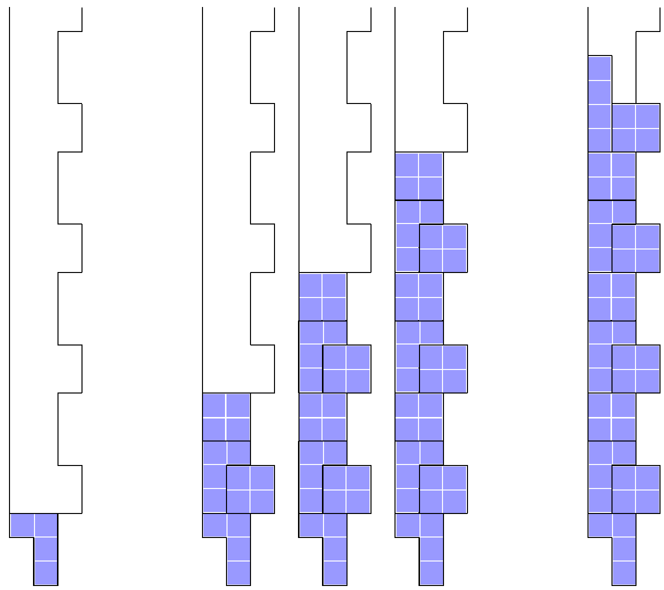
To prove that the reduction is valid we show two things:

1. That a "yes" instance of 3-Partition reduces to an instance of Tetris where the game board can be cleared.
2. That a "no" instance of 3-Partition reduces to an instance of Tetris where there is no possible way to clear the game board.

3.1 A "yes" instance

To fill the game board with a sequence of Tetris pieces arising from a "yes" instance of 3-PARTITION, you do the following:

– The sequence of Tetris pieces up to the 'subset fillers' represents the values $a_i \in A$. As A is part of a "yes" instance it can be divided into s disjoint subsets, all of which have a sum that equals T. To fill the game board you must put the 'values' in the right 'buckets'. To put a 'value' in a 'bucket' you proceed as follows (in this example for $a_i = 3$):



'begin' 'middle' ($a_i = 3$) 'end'

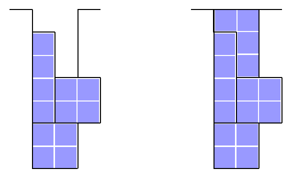
Note that multiple 'values' can be put into one 'bucket' in this fashion, for the shape left after putting one 'value' in the 'bucket' is the same as in the initial situation. Also notice that to put all the values $a_i \in B_j$ in 'bucket' j, the height of a 'bucket' should be

$$\sum_{a_i \in B_j} \{(\text{height of 'begin'}) + a_i \cdot (\text{height of 'middle'}) + (\text{height of 'end'})\}$$

$$= \sum_{a_i \in B_j} \{3 + a_i \cdot 5 + 2\} = \sum_{a_i \in B_j} \{5a_i + 5\}$$

Note that the 'end' and 'begin' overlap in height and that this overlap is counted in the height of 'begin', not in the height of 'end'. Because $\sum_{a_i \in B_j} a_i = T$ and $|B_j| = 3$ for $1 \leq j \leq s$ in a "yes" instance the height turns out to be $5T + 3 \cdot 5 = 5T + 15$, and because the last 'end' sticks out an extra two squares, the total height needed will be $5T + 17$, which is less then H $(= 5T + 18)$ and therefore the height of a 'bucket' is sufficient.

– After filling all the 'buckets' as described above, there will be s buckets which look like (a). These buckets can all be filled to look like (b) using the s 'subset fillers'.



(a) (b)

– Now the whole board is filled except for the 'lock' and the 'fill area'. Next you put the 'lock'-piece in the 'lock'-space and thereby clear the top two lines of the board. Now all that needs to be filled is the 4 by $5T + 16$ rectangular 'fill area' with the $5T + 16$ straight lined Tetris pieces. This can be done by stacking the pieces horizontally and thus you will have cleared the Tetris game board. □

So we have proven that:

Theorem 2. *Using the reduction proposed a "yes" instance of* 3-PARTITION *reduces to an instance of* TETRIS *for which the game board can be cleared.*

3.2 A "no" instance

Using a series of lemmas we will show that a "no" instance of 3-PARTITION, when reduced as proposed in Section 2, results in an instance of TETRIS that is impossible to clear.

Lemma 3. *If there is ever placed a Tetris piece above the bottom* $5T + 18$ *lines then the game board can not be cleared.*

Proof: Every Tetris piece has the same volume and (as proven in Theorem 2) they fill exactly the gaps in the bottom $5T+18$ lines when A is a "yes" instance. The number of Tetris pieces is independent of A being a "yes" or a "no" instance, because $\sum_{a_i \in A} a_i = sT$. If a piece is placed outside the $5T + 18$ lines then more than $5T + 18$ lines need to be cleared and that is simply not possible. Notice that the rules of the game are such that pieces stick to the position where they are left after their initial placement, even though (through line clearing) there may exist empty spaces beneath them. □

Lemma 4. *To be able to clear the game board no other piece than the 'lock'-piece can be placed in the 'lock'-space.*

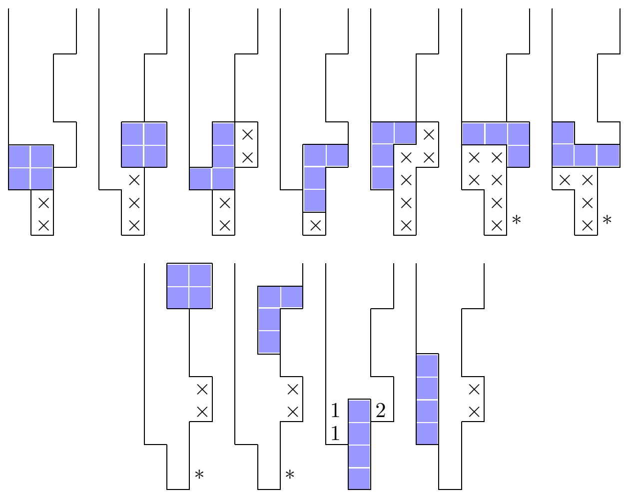
Proof: There is no other piece that would fit into the 'lock'-space without sticking out above the $5T + 18$ lines and then the game board could not be cleared (as proven in Lemma 3). □

Lemma 5. *If placement of a piece preceding the 'lock'-piece ever creates spaces within a 'bucket' that no piece can reach using translation and rotation, then the game board can not be cleared.*

Proof: No lines can be cleared before the 'lock' is filled (clearing the top 2 rows) and this can only be done by the 'lock'-piece (as proven in Lemma 4) and the total volume of the pieces preceding the 'lock'-piece can just fill all the space in the 'buckets'. Therefore, if a space is created within a bucket that can not be filled by translating and rotating pieces, then a volume equal to the space would stick out above the $5T + 18$ lines and therefore the game board can not be cleared (by Lemma 3). □

Lemma 6. *If one of the pieces of a 'value' is put into a different 'bucket' than where the 'begin' piece of this value is put, then the game board can not be cleared.*

Proof: When putting the 'begin' piece of a 'value' into a 'bucket' all the other 'buckets' have the same shape. To prove that no other piece of the 'value' will fit into this 'initial' state of one of the other 'buckets' all possibilities are displayed below:

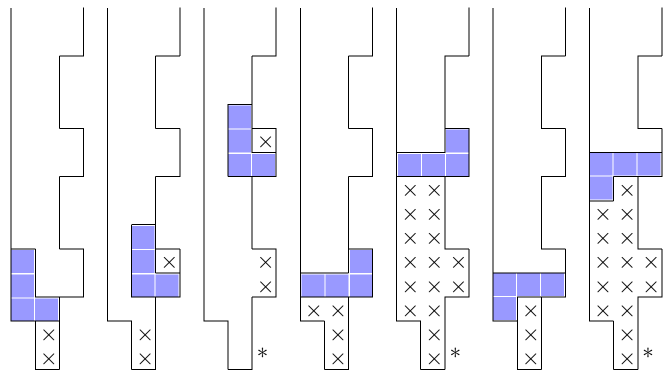


Every $\times$ is a space that is unreachable using translation and rotation. A number stands for a space that can still be reached, but if filled then one of the other numbers becomes unreachable. A '$*$' next to a possibility denotes that the piece can be placed in a higher 'notch' in the same fashion with the same result. Because all options for starting in a 'bucket' with a piece from the 'middle' or the 'end' sequence are shown to contain unreachable spaces and can therefore not be cleared (as proven in Lemma 5), all pieces of a 'value' must be put into one single 'bucket' in order to be able to clear the game board.

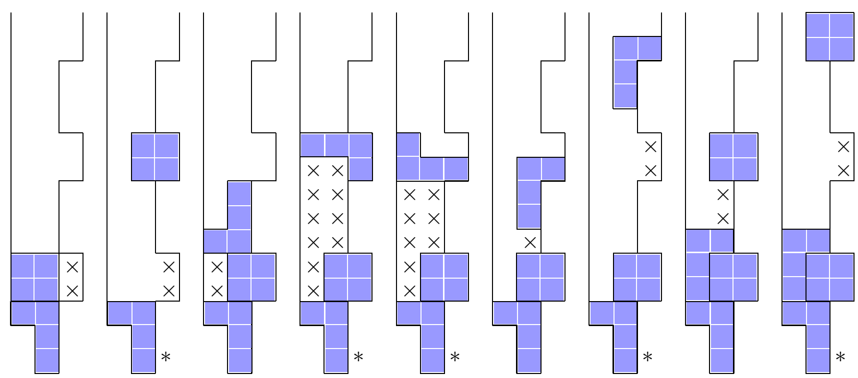
Note that in the sixth possibility the position of the L-shaped piece may seem unlikely. However, the rotation model of Tetris sometimes allows an instantaneous "flip" by 90 degrees, making this possible. □

Lemma 7. *To clear the game board a 'value' should be put in a 'bucket' in exactly the way that is described in Section 3.1.*

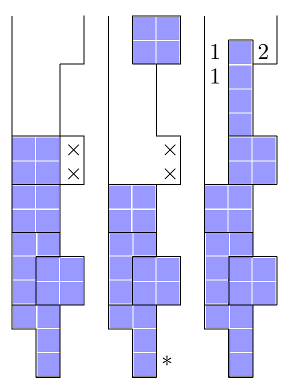
Proof: To prove that there is no other way to put a 'value' in a 'bucket' all other possibilities are displayed below:
For the 'begin' piece:



For the 'middle' pieces:



For the 'end' pieces:



Again every $\times$ is a space that is unreachable using translation and rotation. A number stands for a space that can still be reached, but if filled then one of the other numbers becomes unreachable. A '$*$' next to a possibility denotes that the piece can be placed in a higher 'notch' in the same fashion with the same result. Because all options to put a 'value' in a 'bucket' other than the one proposed in Section 3.1 are shown to contain unreachable spaces and can therefore not be cleared (as proven in Lemma 5), and a 'value' must be put in one single 'bucket' (as proven in Lemma 6) a 'bucket' must be filled in exactly the way described in Section 3.1 in order to clear the game board. □

Lemma 8. *To clear the game board a 'bucket' must contain exactly three 'values' and the sum of these 'values' must be exactly T.*

Proof: There are $T + 3$ 'notches' in every 'bucket'. Because a 'value' must be put into a single 'bucket' (as proven in Lemma 6) and there is only one way to put this 'value' into the 'bucket' (as proven in Lemma 7), a 'value' a_i will always fill $a_i + 1$ 'notches' (a 'notch' for every 'middle' sequence and a 'notch' for the 'begin' and 'end' combined). This means that the number of 'notches' filled in a 'bucket' B is equal to $\sum_{a_i \in B} a_i + |B|$. To clear the game board exactly $T + 3$ 'notches' must be filled in every 'bucket' so $\sum_{a_i \in B} a_i + |B| = T + 3$. If $|B| < 3$ this means that $\sum_{a_i \in B} a_i \geq T + 1$, which is impossible since $a_i < T/2$ for $1 \leq i \leq 3s$. If $|B| > 3$ it would mean that $\sum_{a_i \in B} a_i \leq T - 1$, which is impossible since $T/4 < a_i$ for $1 \leq i \leq 3s$. Thus $|B| = 3$ and $\sum_{a_i \in B} a_i = T$. □

Theorem 9. *Using the reduction proposed a "no" instance of* 3-Partition *reduces to an instance of* Tetris *of which the game board can not be cleared.*

Proof: To clear the game board the s 'buckets' must have exactly three values (as proven in Lemma 8) and the sum of all the 'values' in a 'bucket' must be exactly T (which is also proven in Lemma 8). This can only be the case if A is a "yes" instance. Therefore the game board can not be cleared if A is a "no" instance. □

4 Conclusion

By Theorem 2 and Theorem 9 the NP-complete problem 3-Partition reduces to Tetris, so Tetris is of the same "hardness":

Theorem 10. Tetris *is NP-complete.*

Hence we obtain what Demaine, Hohenberger and Liben-Nowell [2] have proven before us, but we have simplified the task. We defined a smaller initial game board, a smaller and less complex sequence of Tetris pieces and we do not put limitations on the 3-Partition problems that can be reduced. One of the benefits of this approach compared to [2] is that we do not need to prove that the reduction holds for different rotation models. By taking 'buckets' of two columns in width it is virtually impossible to rotate pieces and it reduces the complexity of the proof enormously.

Out of the seven tetrominoes used in Tetris, the construction uses only five. We do not know whether a further reduction in the number of different pieces is possible. It would also be nice to further examine the complexity for different rule sets. For instance, if — as in Cascade Tetris — pieces are allowed to fall further down in later stages (see the proof of Lemma 3), we have a new situation.

References

1. E.D. Demaine, Playing Games with Algorithms: Algorithmic Combinatorial Game Theory, in: J. Sgall, A. Pultr and P. Kolman (editors), *Proceedings 26th International Symposium on Mathematical Foundations of Computer Science MFCS2001*, Lecture Notes in Computer Science 2136, pp. 18–32, 2001.
2. E.D. Demaine, S. Hohenberger and D. Liben-Nowell, *Tetris is Hard, Even to Approximate*, Technical Report MIT-LCS-TR-865 (to appear in COCOON 2003), Laboratory of Computer Science, Massachusetts Institute of Technology, 2002.
3. M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-completeness*, Freeman, 1979.
4. H.J. Hoogeboom and W.A. Kosters, *How to Construct Tetris Configurations*, Technical report Leiden Institute of Advanced Computer Science, Universiteit Leiden, 2003.