

# How Fast Can We Play Tetris Greedily with Rectangular Pieces?

Justin Dallant  

Université libre de Bruxelles, Belgium

John Iacono  

Université libre de Bruxelles, Belgium

---

## Abstract

Consider a variant of Tetris played on a board of width  $w$  and infinite height, where the pieces are axis-aligned rectangles of arbitrary integer dimensions, the pieces can only be moved before letting them drop, and a row does not disappear once it is full. Suppose we want to follow a greedy strategy: let each rectangle fall where it will end up the lowest given the current state of the board. To do so, we want a data structure which can always suggest a greedy move. In other words, we want a data structure which maintains a set of  $O(n)$  rectangles, supports queries which return where to drop the rectangle, and updates which insert a rectangle dropped at a certain position and return the height of the highest point in the updated set of rectangles. We show via a reduction from the Multiphase problem [Pătraşcu, 2010] that on a board of width  $w = \Theta(n)$ , if the OMv conjecture [Henzinger et al., 2015] is true, then both operations cannot be supported in time  $O(n^{1/2-\epsilon})$  simultaneously. The reduction also implies polynomial bounds from the 3-SUM conjecture and the APSP conjecture. On the other hand, we show that there is a data structure supporting both operations in  $O(n^{1/2} \log^{3/2} n)$  time on boards of width  $n^{O(1)}$ , matching the lower bound up to an  $n^{o(1)}$  factor.

**2012 ACM Subject Classification** Theory of computation  $\rightarrow$  Computational geometry; Theory of computation  $\rightarrow$  Problems, reductions and completeness

**Keywords and phrases** Tetris, Fine-grained complexity, Dynamic data structures, Axis-aligned rectangles

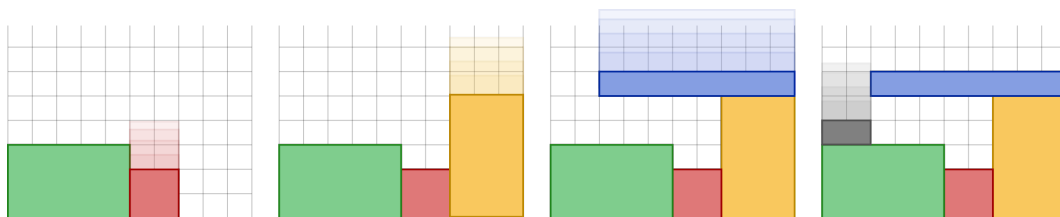
**Digital Object Identifier** 10.4230/LIPIcs.FUN.2022.13


**Related Version** *Full Version:* <https://arxiv.org/abs/2202.10771>

**Funding** *Justin Dallant:* Supported by the French Community of Belgium via the funding of a FRIA grant.

*John Iacono:* Supported by the Fonds de la Recherche Scientifique-FNRS under Grant no MISU F 6001 1.

**Acknowledgements** We thank Nemaou for involuntarily providing us with the initial inspiration for this work.



 **Figure 1** Illustration of a few turns of the game.



© Justin Dallant and John Iacono;  
licensed under Creative Commons License CC-BY 4.0  
11th International Conference on Fun with Algorithms (FUN 2022).

Editors: Pierre Fraigniaud and Yushi Uno; Article No. 13; pp. 13:1–13:19



Leibniz International Proceedings in Informatics

LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

## 1 Introduction

Consider a game played on a board of a certain width  $w$ , which has a wall on the bottom, left and right side but extends on the top as far as we wish. The game goes like this: at each turn, we are given a rectangle of arbitrary integer height and arbitrary integer width (as long as it fits on the board), and we let it drop on the other rectangles (or the bottom of the board) from above starting at the horizontal position of our liking. This is not unlike the basic rule of the famous game Tetris, but instead of having tetrominoes (pieces made of four orthogonally connected unit squares), we have rectangular pieces made of any number of unit squares. See Figure 1 for an example of a possible execution of the game.

Now, our goal as the rectangle droppers is to have the rectangles be as low as possible (in Tetris, once a certain height is reached, it means game over!). There are of course various clever strategies and heuristics which could be adopted here. But assume without loss of plausibility that we (the authors) are greedy, and as greedy players we want to adopt the following strategy: at every turn, we let the rectangle drop at a horizontal position which will minimize its vertical position, with no care for the future rectangles we might be given. The question we seek to answer is: how long do we need to think in each turn when adopting this strategy? Here we assume, with a not-so-insignificant loss of plausibility, that we are as powerful as a word-RAM machine. In particular, all conjectures, cited results and claims in this paper are for the word-RAM model with words of length  $\Theta(\log n)$ , although some carry over to other popular models. All integers appearing in any problem or procedure in this paper have a binary representation of  $O(\log n)$  bits.

Let us formalize this problem with the following definition.

► **Definition 1.** *A Rectangle Dropping Data Structure (RDDS) maintains a set of  $O(n)$  independent<sup>1</sup> axis-aligned rectangles in the plane with integer coordinates, lying on or above the  $x$ -axis and between the vertical lines  $x = 0$  and  $x = w$ , and allows the following:*

- *(Preprocessing) Initialize an empty RDDS containing no rectangle.*
- *(Update) Given an axis-aligned rectangle  $R$  and a non-negative integer  $x_d$ , drop  $R$  with left border at  $x$ -coordinate  $x_d$  (here we assume that  $R$  and  $x_d$  are such that  $R$  will lie between the lines  $x = 0$  and  $x = w$ ).*
- *(Query) Given an axis-aligned rectangle  $R$ , return the position where  $R$  must be dropped to end up as low as possible (or one such position if it is not unique) as well as the height of the highest point in the set of rectangles which would result from that move.*

Note that an RDDS is actually a bit more powerful than what we need to follow the greedy strategy, as this data structure does not force us to play greedily at each turn, but provides a way to suggest a greedy move at every turn. As no one likes to be forced to do anything (even when the thing in question is what we would have done of our own volition), we will study this variant of the problem. Our results could however be adapted to the case where we can only perform updates corresponding to a greedy strategy.

We would like to give some bounds on the time needed to perform the operations of an RDDS. Because in practice it seems hard to prove good lower bounds on the time needed to solve computational problems, one popular approach (dating back to the seminal works of Cook [26] and Karp [54]) has been to show the hardness of some problem assuming the conjectured hardness of some other “key problem” (this constitutes what is called a conditional lower bound). This approach, when dealing with specific polynomial bounds (in

---

<sup>1</sup> We say two rectangles are independent when their interiors do not intersect.

contrast to showing the NP-hardness of a problem for example) is part of a field of study known today as fine-grained complexity (see e.g., introductory surveys by Bringmann [20] and V. V. Williams [94]). Some of these key problems are the 3-SUM problem, All-Pairs-Shortest-Paths (APSP), Boolean Matrix Multiplication (BMM), Triangle Detection in a graph, Boolean Satisfiability (SAT) and the Orthogonal Vectors problem (2OV).

The use of such approaches for data structures solving dynamic problems was pioneered by Pătraşcu [78]. In particular, he introduced the so-called Multiphase problem and showed that it can be fine-grained reduced to many dynamic problems. He also showed that assuming the famous 3-SUM problem is hard, there are polynomial lower bounds to how fast the Multiphase problem can be solved, which in turn implies polynomial lower bounds for the dynamic problems which it reduces to. There has since been much work on conditional lower bounds for dynamic problems [1, 2, 3, 5, 6, 7, 15, 16, 17, 24, 31, 43, 44, 53, 57, 58, 77, 90, 95]. We mention some of the expansions on Pătraşcu’s work relevant to this paper in Section 2.1. In a previous paper by the authors [32], Pătraşcu’s work and these expansions were exploited to obtain conditional polynomial lower bounds for a variety of dynamic geometric problems. The lower bounds obtained here were inspired by this previous work.

After reviewing the Tetris literature in Section 1.1, and the definitions of the hardness conjectures in Section 2, we reduce the Multiphase problem to the conception of an efficient RDDS in Section 3, implying that under the OMv conjecture [43], any RDDS requires essentially  $n^{1/2}$  time per operation. This reduction also implies (weaker, but still polynomial assuming an empty RDDS can be initialized quickly) lower bounds from the 3-SUM conjecture and the APSP conjecture (see Section 2.1 for definitions of these hardness conjectures). We also give in Section 4 a data structure matching the lower bound obtained from the OMv conjecture (up to an  $n^{o(1)}$  factor).

## 1.1 Related work

Tetris, created in 1984 by Alexey Leonidovich Pajitnov of the Dorodnitsyn Computing Centre of the Soviet Academy of Science, was one of the greatest inventions of the USSR, one that was widely embraced by the citizens of both the Warsaw Pact and NATO as well as non-aligned countries. In the computer science literature, the allure of Tetris has been irrepressible, with works using Tetris found in virtually every subdiscipline.

One might be tempted to begin reviewing the literature with the paper whose title is the one single word “Tetris” [96]. However, one will quickly realize that the Tetris that they speak of is not the game so beloved by humanity, but rather “Tetris is an Asynchronous Byzantine Fault Tolerance consensus algorithm designed for next generation high-throughput permission and permissionless blockchain.”

As for the game itself, the most research in the computer science community, unsurprisingly, is about getting computers to play Tetris. There, Tetris has been used as a research tool more often than any other video game [75] and has been subjected to just about every machine learning and artificial intelligence paradigm [4]. However, success is far from guaranteed, as *The Unsuitability of Supervised Backpropagation Networks for Tetris* shows [60].

Researchers have discovered that in addition to computers, humans can also play Tetris. They have studied how humans interact as they play Tetris and how this relates to primate socialization in general, observing that “baboon social intelligence contrasts both with that of Tetris players and human infants, because both have the ability to provoke epistemic effects” [27]. Other human angles have been studied as well, including studying styles of play [18, 55], humanity’s suboptimal choices [80], eye movement while playing [51, 61, 62, 85], and how champion Tetris players play, including the relationship between Hick’s Law and

## 13:4 How Fast Can We Play Tetris Greedily with Rectangular Pieces?

expertise [64], and social exertion [67]. Tetris has been used as well to develop a model of cognition [91]. Additionally, a “Brain Computer Interface (BCI) game that is inspired [by] the Tetris game” was created where they perform experiments with children with attention deficit and hyperactivity disorder (ADHD) [76] (see also [59, 74]).

Several kinds of researchers have worked on helping humans to play Tetris better [68] and addressed the all-important work on the influence of the tempo of the music one listens to while playing Tetris on the score [47, 48, 49]. Other researchers worked to study human performance in Tetris, including the elements that can be used to predict whether a human will become a Tetris master [8, 41, 63, 81, 82]. The issue of dynamically adjusting the difficulty of the game has also been the subject of study [9, 10, 65, 66, 84].

One great step in the development of Tetris was the creation of a mobile version of “the stable and reliable game based on the mature game algorithm” [92]. Also of note was *Pop-out Tetris* [56] where they applied the “Fish Tank Virtual Reality technique to a commodity tablet” as well as an embedded system created for Tetris [72]. The long history of Tetris and its many implementations has led one group to study these implementations over time, deeming them “worthy of historical analysis” [52].

From an education perspective, Tetris was used by [25] to help students learn Chinese proverbs by putting Chinese characters on each Tetris block. Within mainstream computer science education, Tetris is a common subject of programming projects [73].

Tetris has also spawned its own fan fiction, which has been studied: “In [one work] there are also racial and gender complications, the story ending on a humorous note with an absurd twist, as in a Samuel Beckett play, when a message appears over the head of the disillusioned blocks: “Play again?”. It is obvious that in these stories the game of Tetris is an allegory for human life and behavior” [79]. From the point of view of a narrative, “An experiment was made with the game Tetris, showing that a control system designed around evaluation of player’s tension [makes] it possible to obtain an execution trace [close] to a narrative one” [33].

Following the immense success of cannabis legalization, Tetris has followed suit with Enhanced Tetris Legalization [28, 29, 30, 69].

A surprisingly active field has been the ability of a Tetris player to hide messages via gameplay [70, 71, 86, 87].

On the more theoretical side, it has been shown how to view the game through the lens of Krohn-Rhodes theory [50], how to use mixed integer programming to pack Tetris-like pieces in 3D [37], how to create Tetris boards with a given pattern [45], and how to generate polyominoes [38]. In [13], they “propose a straight Coward algorithm to transform a Tetris-like game into its corresponding automaton.”

Closer to the topic of this paper is the algorithmic work relating to the complexity of playing Tetris. The starting point for this consists of results about the hardness of approximating various Tetris objectives (e.g. number of rows cleared, number of moves before reaching a certain height), even with foreknowledge of the entire sequence of pieces [19, 35, 36]. This work was generalized to show NP-completeness for  $k$ -ominoes with  $k \geq 4$ , and surprisingly also show that it is NP-complete to clear the board even for  $k = 2$  when rotations are not allowed [34]. These reductions were further tightened in [11, 12] to when the board is small and in [89] for the variant where you can rotate and then move down, but when no rotations are allowed after the block has begun to fall. The undecidability of whether a set of sequences of Tetris moves described by a regular language contains a sequence that results in an empty board has been proven [46]. While this all sounds very negative, a more positive outlook can be found in *Why Most Decisions Are Easy in Tetris* [83].

Unrelated to Tetris, the problem studied is a form of dynamic *reverse range query* [88]. In a standard range query one has a collection of objects which are preprocessed such that given a range, some question about the objects that intersect the range can be answered. The literature on range queries is immense, with variants depending on the type of objects, the type of ranges, and the specific queries (see, for example, the chapter on range queries in [40]). Classically, what is interesting is that for points and queries that ask how many points intersect the range, the answer is polylog for orthogonal queries and polynomial for non-orthogonal queries, assuming reasonable space. In a reverse range query, we are looking for the ranges that would give a certain result, and to provide the extreme such range under some measure; if there is only a single such reverse range query without parameters, this is interesting as a data structure only when the point set is changing dynamically.

For example, the classic problem of finding the smallest circle enclosing a set of points in the plane can be viewed as such a reverse range query, where one is searching for a circular range query that returns  $n$  points, and among all such queries is looking for the smallest one; in the dynamic case the best known data structure takes polylogarithmic time, even when deletions are allowed [22]. On the other hand, the best known data structure for maintaining the largest empty circle in a set of points (with center constrained to be inside a fixed triangle) takes time  $\tilde{O}(n^{7/8})$  [21] (or  $\tilde{O}(n^{11/12})$  when deletions are also allowed [22]).

For the case of finding (not necessarily axis-aligned) rectangular ranges enclosing all points, the best known data structure takes time  $\tilde{O}(n^{1/2})$  to find the minimum-perimeter range and  $\tilde{O}(n^{5/6})$  for the minimum-area range (also [21]). No conditional lower bounds are known for these problems, which remains a major open problem. In our case we are looking for a three-sided upwards-facing query that returns zero objects,<sup>2</sup> and among all such queries to find the one that is as low as possible.

## 2 A few hardness conjectures

Let us quickly go over some of the key conjectures in fine-grained complexity which will be relevant here. The first, and arguably one of the earliest in the field (with the seminal work of Gajentaan and Overmars [39]), is the 3-SUM conjecture.

► **Definition 2** (3-SUM conjecture). *The following problem (3-SUM) requires  $n^{2-o(1)}$  expected time to solve: given a set of  $n$  integers, decide if three of them sum up to 0.*

The 3-SUM problem can be easily solved in  $O(n^2)$  time, and with a lot of effort this can be slightly improved, both for integer and real inputs [14, 23, 42].

The second problem we will mention is All-Pairs-Shortest-Path (APSP).

► **Definition 3** (APSP conjecture). *The following problem (APSP) requires  $n^{3-o(1)}$  expected time to solve: given a weighted directed graph  $G$  with no negative cycles, compute the distance between every pair of vertices in  $G$ .*

Here, the fastest known algorithm runs in slightly subcubic time [93]. Both problems (3-SUM and APSP) are related in many ways (see e.g., [95]). One particular way in which they are related is that they both reduce in a fine-grained manner to the so-called Exact Triangle problem. In particular, if the following conjecture is false, then both the 3-SUM conjecture and the APSP conjecture are false.

<sup>2</sup> Although we have presented our results as a board containing rectangles, our upper and lower bounds also hold for points.

► **Conjecture 4** (Exact Triangle conjecture). *The following problem (Exact Triangle) requires  $n^{3-o(1)}$  expected time to solve: given a weighted graph  $G$  and a target weight  $T$ , determine if there is a triangle in  $G$  whose edge weights sum to  $T$ .*

The following problem and the corresponding conjecture were introduced by Henzinger et al. [43] as a way to unify most known conditional lower bounds for dynamic problems under a single conjecture, and sometimes even strengthen the known bounds.

► **Definition 5** (Online Boolean Matrix-Vector Multiplication (OMv)). *We are given an  $n \times n$  boolean matrix  $M$ . We can preprocess this matrix, after which we are given a sequence of  $n$  boolean column-vectors of size  $n$  denoted by  $v_1, \dots, v_n$ , one by one. After seeing each vector  $v_i$ , we must output the product  $Mv_i$  before seeing  $v_{i+1}$ .*

► **Conjecture 6** (OMv conjecture). *Solving OMv requires  $n^{3-o(1)}$  expected time.*

## 2.1 Pătraşcu's Multiphase Problem

The following problem was introduced by Pătraşcu [78] as a means to prove polynomial lower bounds for many dynamic problems.

► **Definition 7** (Multiphase Problem). *In the Multiphase Problem, we are given a family  $\mathcal{F} = \{F_1, \dots, F_k\}$  of  $k$  non-empty subsets of  $\{1, 2, \dots, m\}$ , such that every element appears in at least one of the subsets. Let  $n = m \cdot k$ . We want to design a data structure  $D$  which maintains a set of  $O(n)$  objects and allows for the following:*

- (Step 1) First, we read  $\mathcal{F}$  and are allowed  $O(t_1 \cdot n)$  expected time to initialize  $D$ .
- (Step 2) Then, we receive a subset  $J \subset \{1, 2, \dots, m\}$  and are allowed  $O(t_2 \cdot m)$  expected time to modify  $D$ .
- (Step 3) Finally, we are given an index  $1 \leq i \leq k$  and after  $O(t_3)$  expected time we decide if  $J \cap F_i \neq \emptyset$ .

Pătraşcu conjectured that there exist constants  $\gamma > 1$  and  $\delta > 0$  such that if  $k = \Theta(m^\gamma)$  then the Multiphase problem requires  $t_1 + t_2 + t_3 \geq n^\delta$ . While this is still unknown, he showed that this is true if we assume the 3-SUM conjecture. Specifically, he showed the following (taking  $k = \Theta(m^{5/2})$ ).

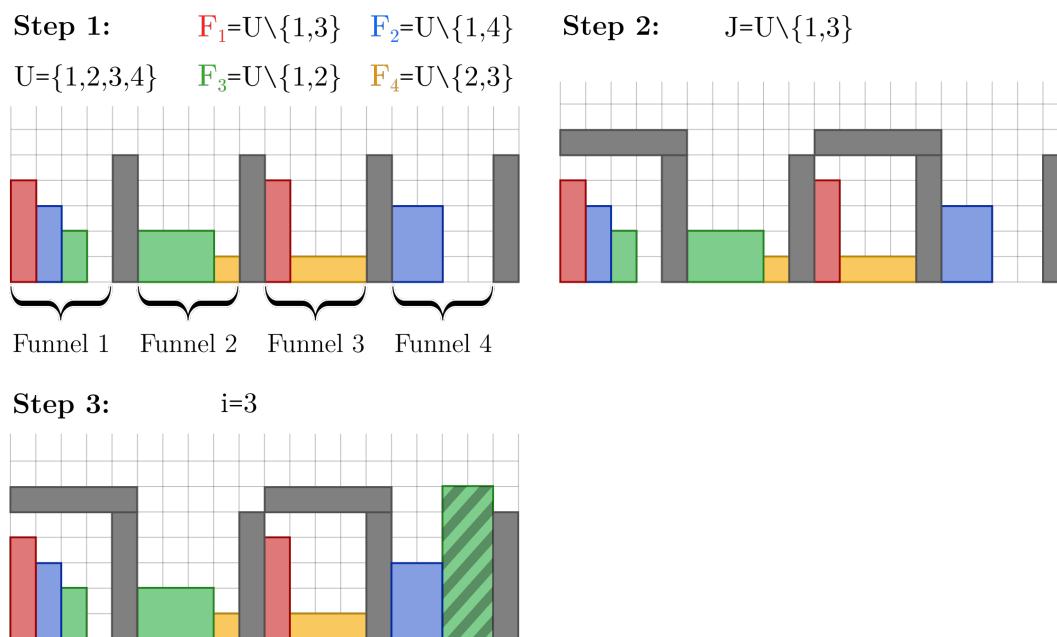
► **Theorem 8** ([78]). *Under the 3-SUM conjecture, the Multiphase Problem requires  $t_1 + t_2 + t_3 \geq n^{1/7-o(1)}$ .*<sup>3</sup>

The techniques of Pătraşcu also allow to have different trade-offs in the lower bounds for  $t_1$ ,  $t_2$  and  $t_3$ . His results have been expanded upon in different ways. On the one hand, Kopelowitz et al.[58] have tightened the reduction from 3-SUM to Set Disjointness (which is an intermediate problem in the reduction from 3-SUM to the Multiphase problem). Their results in particular imply the following:

► **Theorem 9** ([58]). *Under the 3-SUM conjecture, for any  $0 < \gamma < 1$ , the Multiphase Problem requires*

$$t_1 \cdot n + t_2 \cdot n + t_3 \cdot n^{\frac{1+\gamma}{3-2\gamma}} \geq n^{\frac{2}{3-2\gamma}-o(1)}.$$

<sup>3</sup> Note that here we have expressed the bound in terms of  $n = k \cdot m$  whereas Pătraşcu originally expressed his bound in terms of what we call  $m$  in this paper.



■ **Figure 2** Illustration of the reduction from the Multiphase problem to RDDS. Notice that in Step 3, the striped green rectangle can only be dropped to lie at a height lower than 2 in a funnel which is not blocked and where there is no other green rectangle. Such a funnel corresponds to an element of  $U$  which is both in  $J$  and in  $F_3$ .

On the other hand, the hardness of the Multiphase Problem has been shown from other hardness conjectures. Vassilevska Williams and Xu [95] have fine-grained reduced Set Disjointness to the Exact Triangle problem. Their result implies in particular that in the previous theorem, one can replace the 3-SUM conjecture by the Exact Triangle conjecture, giving the following.

► **Theorem 10** ([95]). *Under the Exact Triangle conjecture (and in particular under either the 3-SUM conjecture or the APSP conjecture), for any  $0 < \gamma < 1$ , the Multiphase problem requires*

$$t_1 \cdot n + t_2 \cdot n + t_3 \cdot n^{\frac{1+\gamma}{3-2\gamma}} \geq n^{\frac{2}{3-2\gamma}-o(1)}.$$

Better bounds can be obtained if we assume the OMv conjecture instead. Indeed, the work of Henzinger et al. implies the following.

► **Theorem 11** ([43]). *Under the OMv conjecture, for any  $0 < \gamma < 1$ , if  $t_1$  is at most polynomial in  $n$  then the Multiphase Problem requires*

$$t_2 \cdot n^{1-\gamma} + t_3 \cdot n^\gamma \geq n^{1-o(1)}.$$

### 3 Reduction from the Multiphase Problem

► **Theorem 12.** *If there exists an RDDS which maintains a set of  $O(n)$  for boards of width  $w = \Theta(n)$  with expected preprocessing time  $t_p$ , expected update time  $t_u$  and expected query time  $t_q$ , then we can solve the Multiphase Problem with  $t_1 = t_p/n + t_u$ ,  $t_2 = t_u$  and  $t_3 = t_q$ .*

## 13:8 How Fast Can We Play Tetris Greedily with Rectangular Pieces?

**Proof.** Suppose we have such an RDDS and consider an instance  $\mathcal{F}$  of the Multiphase problem.

We perform Step 1 as follows (see Figure 2). Given a family  $\mathcal{F} = \{F_1, F_2, \dots, F_k\}$  of subsets of  $\{1, 2, \dots, m\}$ , we set the width of the board to  $m \cdot (k + 1) = \Theta(n)$ . For every  $1 \leq j \leq m$ , drop a rectangle of height  $k + 1$  and width 1 with left border at  $x = j(k + 1) - 1$ . This defines  $m$  regions of width  $k$  on the board (between these rectangles and the borders), which we associate with the  $m$  elements of the base set. Now for  $j$  going from 1 to  $m$  we create a sort of “funnel” structure in region  $j$  as follows: for  $i$  going from 1 to  $k$ , if  $j \notin F_i$ , drop a rectangle of height  $k + 1 - i$  such that its right border is at  $x = i + (j - 1)(k + 1)$ , its bottom border is at  $y = 0$  and its left border lies as far left as possible. All in all, we have inserted  $n = O(k \cdot m)$  rectangles in  $O(t_p + n \cdot t_u)$  expected time.

When given  $J \subset \{1, 2, \dots, m\}$  in Step 2, drop a rectangle of width  $k + 1$  and height 1 with left border at  $x = (j - 1)(k + 1)$  for every  $1 \leq j \leq m$  such that  $j \notin J$ . This has the effect of “blocking” the regions which are not relevant in the next step, and costs  $O(m \cdot t_u)$  expected time.

Finally, when given an index  $i$  in Step 3, query the RDDS with a rectangle of width  $k + 1 - i$  and height  $k + 2$ . It is easy to check that if there is some  $j \in J \cap F_i$ , then the bottom of such a rectangle can drop in the funnel corresponding to  $j$  in such a way that its bottom lies at or below  $y = k - i$  (so its top lies at or below  $y = 2k + 2 - i$ ). On the other hand, if  $J \cap F_i = \emptyset$ , then the funnel of every region which was not blocked in the previous is too narrow at height  $y = k + 1 - i$  for the rectangle to fall through. Thus, we can solve the Multiphase Problem with a single query in Step 3, costing  $O(t_q)$  expected time. ◀

This then implies the following:

► **Theorem 13.** *Under the Exact Triangle conjecture (and in particular under either the 3-SUM conjecture or the APSP conjecture), for any  $0 < \gamma < 1$ , an RDDS for boards of width  $w = \Theta(n)$  requires*

$$t_p + t_u \cdot n + t_q \cdot n^{\frac{1+\gamma}{3-2\gamma}} \geq n^{\frac{2}{3-2\gamma}-o(1)}.$$

In particular, for  $\gamma = 2/3$ , we have

$$t_p/n + t_u + t_q \geq n^{1/5-o(1)}.$$

Moreover, if  $t_p$  is at most polynomial in  $n$ , then under the OMv conjecture, for any  $0 < \gamma < 1$  we have

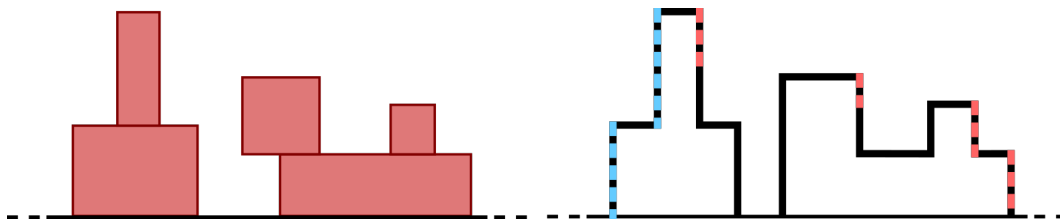
$$t_u \cdot n^{1-\gamma} + t_q \cdot n^\gamma \geq n^{1-o(1)}.$$

In particular, for  $\gamma = 1/2$ , we have

$$t_u + t_q \geq n^{1/2-o(1)}.$$

Note that, as mentioned in the introduction of the paper, with some care this reduction could be adapted to the case where a query is always followed by an update dropping the rectangle at the returned position, and no other updates are allowed (i.e. we are forced to always follow the suggested greedy strategy). This can be done by creating “notches” of different widths and heights at the top of the rectangles delimiting the different funnels, so that in Step 2 the funnel over which a rectangle is dropped can be chosen by specifying the width of the rectangle. In particular we could get the same bound from the OMv conjecture





■ **Figure 3** Illustration of the skyline, left staircase (blue dotted line) and right staircase (red dotted line) of a set of rectangles.

(exploiting the fact that Theorem 11 already holds for the Multiphase problem where  $k = m$ ). It could also be adapted to the case where we are allowed to rotate the rectangles before dropping them, by stretching all the rectangles enough horizontally so that rotating them is never advantageous (here we would need to consider boards of larger than linear width). As rectangles are a special case of polyominoes, all these reductions hold also for arbitrary polyominoes of size  $O(n)$ .

#### 4 A near optimal data structure

The previous section shows that if we want to follow the greedy strategy, then we cannot always make a move very quickly. While this at least gives us a pretty good excuse for our slow plays, let us now focus on the positive side and prove that we can match this lower bound up to an  $n^{o(1)}$  factor.

► **Definition 14.** *Given a finite set  $S$  of axis-aligned rectangles in the plane lying on or above  $x$ -axis, the skyline of  $S$  is the union of the top, left and right boundaries of the region  $B$  obtained as follows:*

- extend the bottom of every rectangle in  $S$  so that its bottom lies on the  $x$ -axis;
- let  $B$  be the union of all these newly-obtained rectangles.

*The left staircase of  $S$  consists of the parts of the skyline of  $S$  visible from the point at  $(-\infty, 0)$ . Similarly, the right staircase of  $S$  consists of the parts of the skyline of  $S$  visible from the point at  $(+\infty, 0)$ .*

Notice that starting with a set of  $n$  independent axis-aligned rectangles, its skyline is a set of curves consisting of at most  $4n - 1$  axis-aligned segments, while the left and right staircases each consist of at most  $n$  vertical segments. Moreover, the skyline contains all the information needed to know how far we can drop a rectangle of a certain width.

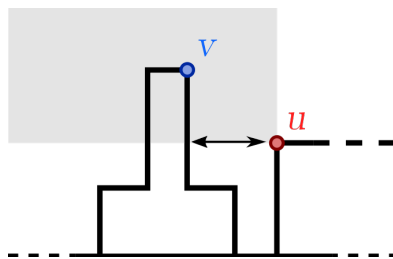
To get an RDDS with  $n^{1/2}$  update and query time, up to a polylogarithmic factor, we will maintain the skyline, left and right staircases of our set of rectangles, broken into at most  $O(n^{1/2})$  chunks each consisting of a sequence of roughly  $n^{1/2}$  segments appearing consecutively on the skyline.

To start, it is an easy exercise to show the following via a simple plane sweep.

► **Lemma 15.** *Given a set of  $n$  independent axis-aligned rectangles, one can construct its skyline, left and right staircase in  $O(n \log n)$  time (where we store the segments appearing in each in order of appearance left to right in three distinct arrays).*

Now, instead of looking for how low we can drop a rectangle of a given width, let us instead turn this question on its head and ask how wide of a rectangle we can drop if we want it to lie at or below a certain height  $h$ . If this is impossible, let's say this maximum width is

## 13:10 How Fast Can We Play Tetris Greedily with Rectangular Pieces?



■ **Figure 4** The width of the largest rectangle whose right border touches  $u$  is the horizontal distance from  $u$  to the horizontally nearest vertex in the shaded region (which in this case is  $v$ ).

0. Remember that we are still constrained by two vertical lines between which our rectangle needs to lie (otherwise, we could have our rectangle be as wide as we wish and simply drop it next to the existing rectangles). Two special cases arise: when the widest rectangle which can be dropped to height  $h$  or below touches the left vertical line or when this rectangle touches the right vertical line. The maximum width of a rectangle corresponding to these cases can be determined in  $O(\log n)$  time given the left and right staircase of the existing rectangles, simply by binary searching through both and finding where they meet the horizontal line  $y = h$ . What about the general case? Here we have the following.

► **Lemma 16.** *Given the skyline of a set of  $n$  rectangles, we can construct a data structure in  $O(n \log n)$  with which given some height  $h$ , we can return the width and position of the widest rectangle which can be dropped at or below height  $h$  in  $O(\log n)$  time.*

**Proof.** We will precompute the answer for all heights  $h$  which correspond to some vertex in the skyline.

We start by computing for every height  $h$ , the widest rectangle which can be dropped so that its right-side touches a vertex at height  $h$  on the skyline. We do this by traversing the skyline from left to right and maintaining the answer for the visited vertices in some data-structure allowing for  $O(\log n)$  lookup and updates (some flavor of self-balancing binary search tree for example). When we reach some vertex  $u$  at height  $h$ , we want to quickly determine how wide we can make a rectangle whose lower-right corner touches  $u$  without it intersecting the interior of the skyline. Notice that this is the difference in  $x$ -coordinates between  $u$  and the horizontally nearest neighbor to the left of  $u$  which is strictly above  $u$  (see Figure 4). If we store the previously visited vertices in a self-balancing binary search tree sorted by  $y$ -coordinate and store in every node of that tree the rightmost vertex in the corresponding subtree, we can get this information in  $O(\log n)$  time. If this width is larger than the current largest width stored for height  $h$ , update it. Overall this procedure can be carried out in  $O(n \log n)$  time.

Similarly by traversing the skyline from right to left we compute for every height  $h$ , the widest rectangle which can be dropped so that its left-side touches a vertex at height  $h$  on the skyline.

Then, for every height  $h$  corresponding to some vertex, the widest rectangle that can drop at or below height  $h$  is the maximum width of a rectangle touching a vertex at height  $h'$  for all  $h' \leq h$ . We can compute this for all relevant  $h$ , as well as the corresponding location where a rectangle should be dropped, in linear time given the information we have already computed. We store this information in a binary search tree ordered by  $h$ . Now given any height (not necessarily corresponding to a vertex of the skyline), we can determine in  $O(\log n)$  how wide a rectangle can be dropped at or below that height (and where to drop it) by looking in the binary search tree for the largest stored height no larger than  $h$ . ◀

We are now ready to break the skyline into chunks to make a data structure which supports insertions.

► **Theorem 17.** *There is a data structure which maintains a set of  $O(n)$  independent axis-aligned rectangles in the plane, lying on or above the  $x$  and between the vertical lines  $x = 0$  and  $x = w$ , with  $O(n \log n)$  construction time and which allows the following operations:*

- (Update) *Given an axis-aligned rectangle  $R$  and a number  $x_d$ , drop  $R$  at  $x$ -coordinate  $x_d$  (here we assume that  $R$  and  $x_d$  are such that  $R$  will lie between the lines  $x = 0$  and  $x = w$ ).*
- (Query) *Given a height  $h$  return the widest rectangle which can be dropped to lie at height  $h$ , together with the position at which it needs to be dropped.*

*Updates can be supported in  $O(n^{1/2} \log^{3/2} n)$  time and queries in  $O(n^{1/2} \log^{1/2} n)$  time.*

**Proof.** To build our data structure, we start by computing the skyline of the set of rectangles in  $O(n \log n)$  time. We add to the skyline long vertical segments corresponding to the lines  $x = 0$  and  $x = w$ . Then we break the skyline into  $O((n/\log n)^{1/2})$  contiguous pieces, each consisting of a skyline having at most  $2(n \log n)^{1/2}$  vertices. We also make sure that the number of vertices in any two consecutive chunks of skyline sums up to at least  $(n \log n)^{1/2}$ . For each of those, we build the data structure of Lemma 16, the left and right staircase and compute the maximum height of a vertex. We store these data structures in a pointer list  $L$ , in left to right order. This costs a total time of  $O((n/\log n)^{1/2} \cdot (n \log n)^{1/2} \log n) = O(n \log n)$ .

Now, given a query height  $h$ , we proceed as follows. We query each of the small data structures in  $L$  to find the maximum width of a rectangle which can be dropped at or below height  $h$  over all of them. This costs  $O((n/\log n)^{1/2} \log n) = O((n \log n)^{1/2})$  time. We are left to deal with rectangles which could span across multiple of these skyline chunks. Here we traverse  $L$  with two pointers  $p_1$  and  $p_2$  in a “sliding window” fashion:

- We start with  $p_1$  at the head of the list and  $p_2$  its successor. While both have not reached the end of the list, we go through the following steps.
- If the maximum height of a vertex in the skyline chunk corresponding to  $p_2$  is at or below  $h$  and  $p_2$  has not yet reached the tail of the list, we move  $p_2$  to its successor in  $L$ .
- Otherwise, we find the widest rectangle which can be dropped at or below height  $h$  and overlaps both skylines corresponding to  $p_1$  and  $p_2$  partially by binary searching through the right staircase corresponding to  $p_1$  and the left staircase of  $p_2$ . Once this is done, we move  $p_1$  over to  $p_2$  and move  $p_2$  to its successor (if  $p_2$  has not yet reached the tail of the list).

By keeping track of the maximum width and the position it was encountered at throughout this procedure, they can be reported in  $O((n/\log n)^{1/2} \log n) = O(n^{1/2} \log^{1/2} n)$  time.

Now, given a rectangle  $R$  to drop at a given position, we first find all chunks of skyline in  $L$  which will be completely covered by  $R$  as well as the pointers  $p_1$  and  $p_2$  to the elements in  $L$  corresponding to the skylines which overlap with the left and right border of  $R$ . By binary searching through the corresponding staircases and finding the maximum height of a vertex in the chunks which are covered by  $R$  we can find out at which height it will end up. Notice that the skylines which are covered by  $R$  are no longer relevant at this point, so we can ignore the corresponding data structures in  $L$  by moving the successor pointer of  $p_1$  so that it points to  $p_2$ . We now rebuild the data structure  $p_1$  points to, including the rectangle  $R$ , as well as the one  $p_2$  points to, excluding all parts of the skyline covered by  $R$ . If this causes any of the two skylines to have more than  $2(n \log n)^{1/2}$  vertices, we can simply divide it in approximate halves and rebuild the two corresponding data structures from scratch in  $O((n \log n)^{1/2} \cdot \log n)$  time each. If two consecutive skylines have fewer than  $(n \log n)^{1/2}$  vertices together, merge

## 13:12 How Fast Can We Play Tetris Greedily with Rectangular Pieces?

them and rebuild the corresponding data structure from scratch in  $O((n \log n)^{1/2} \cdot \log n)$  time. These divisions and merges can happen at most twice for a given update, and ensure that we always have  $O((n/\log n)^{1/2})$  small data structures storing  $O((n \log n)^{1/2})$  vertices each. The total cost of an update is then  $O((n/\log n)^{1/2} + (n \log n)^{1/2} \cdot \log n) = O(n^{1/2} \log^{3/2} n)$ . ◀

Given such a data structure, we can solve our original question. We keep track of all the  $O(n)$  possible heights a rectangle could be dropped at, and when given a rectangle of a certain width, we do a binary search through all possible heights to find the lowest height which will accommodate a rectangle of that width. This increases the query time by a factor of  $O(\log n)$ , giving the following:

► **Theorem 18.** *There is an RDDS with  $O(n \log n)$  construction time which supports queries and updates in  $O(n^{1/2} \log^{3/2} n)$  time.*

This result applies even in the case where the RDDS is not necessarily initialized with an empty set of rectangles. Considering the lower bound from the OMv conjecture given in the previous section, this is likely close to optimal (although we have not made a big effort to optimize the polylogarithmic terms in the runtime). If one were to face an opponent playing the game with a sub-polynomial speed advantage over this data structure, we venture to guess this could be mitigated by an appropriate amount of distraction and foul play.

---

### References

- 1 Amir Abboud and Søren Dahlgaard. Popular conjectures as a barrier for dynamic planar graph algorithms. In Irit Dinur, editor, *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 477–486. IEEE Computer Society, 2016. doi:10.1109/FOCS.2016.58.
- 2 Amir Abboud and Virginia Vassilevska Williams. Popular conjectures imply strong lower bounds for dynamic problems. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 434–443. IEEE Computer Society, 2014. doi:10.1109/FOCS.2014.53.
- 3 Amir Abboud, Virginia Vassilevska Williams, and Huacheng Yu. Matching triangles and basing hardness on an extremely popular conjecture. *SIAM J. Comput.*, 47(3):1098–1122, 2018. doi:10.1137/15M1050987.
- 4 Simón Algorta and Özgür Simsek. The game of Tetris in machine learning. *CoRR*, abs/1905.01652, 2019. arXiv:1905.01652.
- 5 Josh Alman, Matthias Mních, and Virginia Vassilevska Williams. Dynamic parameterized problems and algorithms. In Ioannis Chatzigiannakis, Piotr Indyk, Fabian Kuhn, and Anca Muscholl, editors, *44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, July 10-14, 2017, Warsaw, Poland*, volume 80 of *LIPICs*, pages 41:1–41:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPICs.ICALP.2017.41.
- 6 Amihod Amir, Timothy M. Chan, Moshe Lewenstein, and Noa Lewenstein. On hardness of jumbled indexing. In Javier Esparza, Pierre Fraigniaud, Thore Husfeldt, and Elias Koutsoupias, editors, *Automata, Languages, and Programming - 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part I*, volume 8572 of *Lecture Notes in Computer Science*, pages 114–125. Springer, 2014. doi:10.1007/978-3-662-43948-7\_10.
- 7 Amihod Amir, Tsvi Kopelowitz, Avivit Levy, Seth Pettie, Ely Porat, and B. Riva Shalom. Mind the gap! - online dictionary matching with one gap. *Algorithmica*, 81(6):2123–2157, 2019. doi:10.1007/s00453-018-0526-2.
- 8 Diana Sofía Lora Ariza. El clustering de jugadores de Tetris. In David Camacho, Marco Antonio Gómez-Martín, and Pedro Antonio González-Calero, editors, *Proceedings 2st Congreso de la Sociedad Española para las Ciencias del Videojuego, Barcelona, Spain, June 24, 2015*,

- volume 1394 of *CEUR Workshop Proceedings*, pages 36–45. CEUR-WS.org, 2015. URL: [http://ceur-ws.org/Vol-1394/paper\\_4.pdf](http://ceur-ws.org/Vol-1394/paper_4.pdf).
- 9 Diana Sofía Lora Ariza, Antonio A. Sánchez-Ruiz, and Pedro A. González-Calero. Time series and case-based reasoning for an intelligent Tetris game. In David W. Aha and Jean Lieber, editors, *Case-Based Reasoning Research and Development - 25th International Conference, ICCBR 2017, Trondheim, Norway, June 26-28, 2017, Proceedings*, volume 10339 of *Lecture Notes in Computer Science*, pages 185–199. Springer, 2017. doi:10.1007/978-3-319-61030-6\_13.
  - 10 Diana Sofía Lora Ariza, Antonio A. Sánchez-Ruiz, and Pedro A. González-Calero. Towards finding flow in Tetris. In Kerstin Bach and Cindy Marling, editors, *Case-Based Reasoning Research and Development - 27th International Conference, ICCBR 2019, Otzenhausen, Germany, September 8-12, 2019, Proceedings*, volume 11680 of *Lecture Notes in Computer Science*, pages 266–280. Springer, 2019. doi:10.1007/978-3-030-29249-2\_18.
  - 11 Sualeh Asif, Michael J. Coulombe, Erik D. Demaine, Martin L. Demaine, Adam Hesterberg, Jayson Lynch, and Mihir Singhal. Tetris is np-hard even with  $O(1)$  rows or columns. *J. Inf. Process.*, 28:942–958, 2020. doi:10.2197/ipsjip.28.942.
  - 12 Sualeh Asif, Michael J. Coulombe, Erik D. Demaine, Martin L. Demaine, Adam Hesterberg, Jayson Lynch, and Mihir Singhal. Tetris is np-hard even with  $O(1)$  rows or columns. *CoRR*, abs/2009.14336, 2020. arXiv:2009.14336.
  - 13 Davide Baccherini and Donatella Merlini. Combinatorial analysis of Tetris-like games. *Discret. Math.*, 308(18):4165–4176, 2008. doi:10.1016/j.disc.2007.08.009.
  - 14 Ilya Baran, Erik D. Demaine, and Mihai Patrascu. Subquadratic algorithms for 3sum. *Algorithmica*, 50(4):584–596, 2008. doi:10.1007/s00453-007-9036-3.
  - 15 Surender Baswana, Shreejit Ray Chaudhury, Keerti Choudhary, and Shahbaz Khan. Dynamic DFS in undirected graphs: breaking the  $o(m)$  barrier. In Robert Krauthgamer, editor, *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 730–739. SIAM, 2016. doi:10.1137/1.9781611974331.ch52.
  - 16 Christoph Berkholz, Jens Keppeler, and Nicole Schweikardt. Answering conjunctive queries under updates. In Emanuel Sallinger, Jan Van den Bussche, and Floris Geerts, editors, *Proceedings of the 36th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS 2017, Chicago, IL, USA, May 14-19, 2017*, pages 303–318. ACM, 2017. doi:10.1145/3034786.3034789.
  - 17 Christoph Berkholz, Jens Keppeler, and Nicole Schweikardt. Answering UCQs under updates and in the presence of integrity constraints. In Benny Kimelfeld and Yael Amsterdamer, editors, *21st International Conference on Database Theory, ICDT 2018, March 26-29, 2018, Vienna, Austria*, volume 98 of *LIPICs*, pages 8:1–8:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPICs.ICDT.2018.8.
  - 18 Jacquelyn H. Berry and Wayne D. Gray. HOT: higher order Tetris, experts’ subgoals and activities. In Ashok K. Goel, Colleen M. Seifert, and Christian Freksa, editors, *Proceedings of the 41th Annual Meeting of the Cognitive Science Society, CogSci 2019: Creativity + Cognition + Computation, Montreal, Canada, July 24-27, 2019*, page 3409. cognitivesciencesociety.org, 2019. URL: <https://mindmodeling.org/cogsci2019/papers/0717/index.html>.
  - 19 Ron Breukelaar, Erik D. Demaine, Susan Hohenberger, Hendrik Jan Hoogetboom, Walter A. Kosters, and David Liben-Nowell. Tetris is hard, even to approximate. *Int. J. Comput. Geom. Appl.*, 14(1-2):41–68, 2004. doi:10.1142/S0218195904001354.
  - 20 Karl Bringmann. Fine-grained complexity theory (tutorial). In Rolf Niedermeier and Christophe Paul, editors, *36th International Symposium on Theoretical Aspects of Computer Science, STACS 2019, March 13-16, 2019, Berlin, Germany*, volume 126 of *LIPICs*, pages 4:1–4:7. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPICs.STACS.2019.4.
  - 21 Timothy M. Chan. Semi-online maintenance of geometric optima and measures. *SIAM J. Comput.*, 32(3):700–716, 2003. doi:10.1137/S0097539702404389.

## 13:14 How Fast Can We Play Tetris Greedily with Rectangular Pieces?

- 22 Timothy M. Chan. Dynamic geometric data structures via shallow cuttings. *Discret. Comput. Geom.*, 64(4):1235–1252, 2020. doi:10.1007/s00454-020-00229-5.
- 23 Timothy M. Chan. More logarithmic-factor speedups for 3sum, (median, +)-convolution, and some geometric 3sum-hard problems. *ACM Trans. Algorithms*, 16(1):7:1–7:23, 2020. doi:10.1145/3363541.
- 24 Lijie Chen, Erik D. Demaine, Yuzhou Gu, Virginia Vassilevska Williams, Yinzhan Xu, and Yuancheng Yu. Nearly optimal separation between partially and fully retroactive data structures. In David Eppstein, editor, *16th Scandinavian Symposium and Workshops on Algorithm Theory, SWAT 2018, June 18-20, 2018, Malmö, Sweden*, volume 101 of *LIPICs*, pages 33:1–33:12. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPICs.SWAT.2018.33.
- 25 Tsung Teng Chen. Chitetris: A Chinese proverb learning game utilizing Tetris game plot. In Gautam Biswas, Diane Carr, Yam San Chee, and Wu-Yuin Hwang, editors, *2010 Third IEEE International Conference on Digital Game and Intelligent Toy Enhanced Learning, DIGITEL 2010, Kaohsiung, Taiwan, April 12-16, 2010*, pages 194–197. IEEE, 2010. doi:10.1109/DIGITEL.2010.34.
- 26 Stephen A. Cook. The complexity of theorem-proving procedures. In Michael A. Harrison, Ranan B. Banerji, and Jeffrey D. Ullman, editors, *Proceedings of the 3rd Annual ACM Symposium on Theory of Computing, May 3-5, 1971, Shaker Heights, Ohio, USA*, pages 151–158. ACM, 1971. doi:10.1145/800157.805047.
- 27 Stephen J. Cowley and Karl F. MacDorman. What baboons, babies and Tetris players tell us about interaction: a biosocial view of norm-based social learning. *Connect. Sci.*, 18(4):363–378, 2006. doi:10.1080/09540090600879703.
- 28 Antonios N. Dadaliaris, Evangelia Nerantzaki, Maria G. Koziri, Panagiotis Oikonomou, Thanasis Loukopoulos, and Georgios I. Stamoulis. Performance evaluation of Tetris-based legalization heuristics. In *Proceedings of the 20th Pan-Hellenic Conference on Informatics, Patras, Greece, November 10-12, 2016*, page 60. ACM, 2016. doi:10.1145/3003733.3003778.
- 29 Antonios N. Dadaliaris, Evangelia Nerantzaki, Panagiotis Oikonomou, Yannis Hatzaras, Angeliki-Olympia Troumpoulou, Ioannis Arvanitakis, and Georgios I. Stamoulis. Enhanced Tetris legalization. In *Proceedings of the SouthEast European Design Automation, Computer Engineering, Computer Networks and Social Media Conference, SEEDA-CECNM 2016, Kastoria, Greece, September 25-27, 2016*, pages 32–35. ACM, 2016. doi:10.1145/2984393.2984410.
- 30 Antonios N. Dadaliaris, Panagiotis Oikonomou, Maria G. Koziri, Evangelia Nerantzaki, Yannis Hatzaras, Dimitrios Garyfallou, Thanasis Loukopoulos, and Georgios I. Stamoulis. Heuristics to augment the performance of Tetris legalization: Making a fast but inferior method competitive. *J. Low Power Electron.*, 13(2):220–230, 2017. doi:10.1166/jolpe.2017.1483.
- 31 Søren Dahlgaard. On the hardness of partially dynamic graph problems and connections to diameter. In Ioannis Chatzigiannakis, Michael Mitzenmacher, Yuval Rabani, and Davide Sangiorgi, editors, *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, volume 55 of *LIPICs*, pages 48:1–48:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPICs.ICALP.2016.48.
- 32 Justin Dallant and John Iacono. Conditional lower bounds for dynamic geometric measure problems, 2021. arXiv:2112.10095.
- 33 Guylain Delmas, Ronan Champagnat, and Michel Augeraud. Plot control for emergent narrative: A case study on Tetris. *Int. J. Intell. Games Simul.*, 5(1), 2008. URL: [http://www.scit.wlv.ac.uk/OJS\\_IJIGS/index.php/IJIGS/article/view/56](http://www.scit.wlv.ac.uk/OJS_IJIGS/index.php/IJIGS/article/view/56).
- 34 Erik D. Demaine, Martin L. Demaine, Sarah Eisenstat, Adam Hesterberg, Andrea Lincoln, Jayson Lynch, and Yun William Yu. Total Tetris: Tetris with monominoes, dominoes, trominoes, pentominoes, . *J. Inf. Process.*, 25:515–527, 2017. doi:10.2197/ipsjjip.25.515.
- 35 Erik D. Demaine, Susan Hohenberger, and David Liben-Nowell. Tetris is hard, even to approximate. *CoRR*, cs.CC/0210020, 2002. URL: <https://arxiv.org/abs/cs/0210020>.

- 36 Erik D. Demaine, Susan Hohenberger, and David Liben-Nowell. Tetris is hard, even to approximate. In Tandy J. Warnow and Binhai Zhu, editors, *Computing and Combinatorics, 9th Annual International Conference, COCOON 2003, Big Sky, MT, USA, July 25-28, 2003, Proceedings*, volume 2697 of *Lecture Notes in Computer Science*, pages 351–363. Springer, 2003. doi:10.1007/3-540-45071-8\_36.
- 37 Giorgio Fasano. A MIP approach for some practical packing problems: Balancing constraints and tetris-like items. *4OR*, 2(2):161–174, 2004. doi:10.1007/s10288-004-0037-7.
- 38 Enrico Formenti and Paolo Massazza. From Tetris to polyominoes generation. *Electron. Notes Discret. Math.*, 59:79–98, 2017. doi:10.1016/j.endm.2017.05.007.
- 39 Anka Gajentaan and Mark H. Overmars. On a class of  $o(n^2)$  problems in computational geometry. *Comput. Geom.*, 5:165–185, 1995. doi:10.1016/0925-7721(95)00022-2.
- 40 Jacob E. Goodman and Joseph O’Rourke, editors. *Handbook of Discrete and Computational Geometry, Second Edition*. Chapman and Hall/CRC, 2004. doi:10.1201/9781420035315.
- 41 Wayne D. Gray, Ray S. Perez, John K. Lindstedt, Anna Skinner, Robin Johnson, Richard E. Mayer, Deanne Adams, and Robert K. Atkinson. Tetris as research paradigm: An approach to studying complex cognitive skills. In Paul Bello, Marcello Guarini, Marjorie McShane, and Brian Scassellati, editors, *Proceedings of the 36th Annual Meeting of the Cognitive Science Society, CogSci 2014, Quebec City, Canada, July 23-26, 2014*. cognitivesciencesociety.org, 2014. URL: <https://mindmodeling.org/cogsci2014/papers/019/>.
- 42 Allan Grønlund and Seth Pettie. Threesomes, degenerates, and love triangles. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 621–630. IEEE Computer Society, 2014. doi:10.1109/FOCS.2014.72.
- 43 Monika Henzinger, Sebastian Krinninger, Danupon Nanongkai, and Thatchaphol Saranurak. Unifying and strengthening hardness for dynamic problems via the online matrix-vector multiplication conjecture. In Rocco A. Servedio and Ronitt Rubinfeld, editors, *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 21–30. ACM, 2015. doi:10.1145/2746539.2746609.
- 44 Monika Henzinger, Andrea Lincoln, Stefan Neumann, and Virginia Vassilevska Williams. Conditional hardness for sensitivity problems. In Christos H. Papadimitriou, editor, *8th Innovations in Theoretical Computer Science Conference, ITCS 2017, January 9-11, 2017, Berkeley, CA, USA*, volume 67 of *LIPICs*, pages 26:1–26:31. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPICs.ITCS.2017.26.
- 45 Hendrik Jan Hoogeboom and Walter A. Kosters. How to construct Tetris configurations. *Int. J. Intell. Games Simul.*, 3(2):97–105, 2004. URL: [http://www.scit.wlv.ac.uk/OJS\\_IJIGS/index.php/IJIGS/article/view/95](http://www.scit.wlv.ac.uk/OJS_IJIGS/index.php/IJIGS/article/view/95).
- 46 Hendrik Jan Hoogeboom and Walter A. Kosters. Tetris and decidability. *Inf. Process. Lett.*, 89(6):267–272, 2004. doi:10.1016/j.ipl.2003.12.006.
- 47 Aline Hufschmitt, Stéphane Cardon, and Éric Jacopin. Can musical tempo makes Tetris game harder? In *IEEE Conference on Games, CoG 2020, Osaka, Japan, August 24-27, 2020*, pages 608–611. IEEE, 2020. doi:10.1109/CoG47356.2020.9231593.
- 48 Aline Hufschmitt, Stéphane Cardon, and Éric Jacopin. Manipulating player performance via music tempo in Tetris. In Pejman Mirza-Babaei, Victoria McArthur, Vero Vanden Abeele, and Max Birk, editors, *CHI PLAY ’20: The Annual Symposium on Computer-Human Interaction in Play, Virtual Event, Canada, November 2-4, 2020 - Companion Volume / Extended Abstracts*, pages 146–152. ACM, 2020. doi:10.1145/3383668.3419934.
- 49 Aline Hufschmitt, Stéphane Cardon, and Éric Jacopin. Dynamic manipulation of player performance with music tempo in Tetris. In Tracy Hammond, Katrien Verbert, Dennis Parra, Bart P. Knijnenburg, John O’Donovan, and Paul Teale, editors, *IUI ’21: 26th International Conference on Intelligent User Interfaces, College Station, TX, USA, April 13-17, 2021*, pages 290–296. ACM, 2021. doi:10.1145/3397481.3450684.

## 13:16 How Fast Can We Play Tetris Greedily with Rectangular Pieces?

- 50 Peter C. Jentsch and Chrystopher L. Nehaniv. Exploring Tetris as a transformation semigroup. *CoRR*, abs/2004.09022, 2020. [arXiv:2004.09022](https://arxiv.org/abs/2004.09022).
- 51 Patrick Jermann, Marc-Antoine Nüssli, and Weifeng Li. Using dual eye-tracking to unveil coordination and expertise in collaborative Tetris. In Tom McEwan and Lachlan McKinnon, editors, *Proceedings of the 2010 British Computer Society Conference on Human-Computer Interaction, BCS-HCI 2010, Dundee, United Kingdom, 6-10 September 2010*, pages 36–44. ACM, 2010. URL: <http://dl.acm.org/citation.cfm?id=2146309>.
- 52 Will Jordan. Morphology of the tetromino-stacking game: The design evolution of Tetris [abstract]. In Tanya Krzywinska, Helen W. Kennedy, and Barry Atkins, editors, *Proceedings of the 2009 DiGRA International Conference: Breaking New Ground: Innovation in Games, Play, Practice and Theory, DiGRA 2009, London, UK, September 1-4, 2009*. Digital Games Research Association, 2009. URL: <http://www.digra.org/digital-library/publications/morphology-of-the-tetromino-stacking-game-the-design-evolution-of-tetris-abstract/>.
- 53 Adam Karczmarz and Jakub Lacki. Fast and simple connectivity in graph timelines. In Frank Dehne, Jörg-Rüdiger Sack, and Ulrike Stege, editors, *Algorithms and Data Structures - 14th International Symposium, WADS 2015, Victoria, BC, Canada, August 5-7, 2015. Proceedings*, volume 9214 of *Lecture Notes in Computer Science*, pages 458–469. Springer, 2015. doi:10.1007/978-3-319-21840-3\_38.
- 54 Richard M. Karp. Reducibility among combinatorial problems. In Raymond E. Miller and James W. Thatcher, editors, *Proceedings of a symposium on the Complexity of Computer Computations, held March 20-22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, USA*, The IBM Research Symposia Series, pages 85–103. Plenum Press, New York, 1972. doi:10.1007/978-1-4684-2001-2\_9.
- 55 Stephen A. Keating, Wan-Chen Lee, Travis W. Windleharth, and Lee Jin Ha. The style of Tetris is...possibly Tetris?: Creative professionals’ description of video game visual styles. In Tung Bui, editor, *50th Hawaii International Conference on System Sciences, HICSS 2017, Hilton Waikoloa Village, Hawaii, USA, January 4-7, 2017*, pages 1–10. ScholarSpace / AIS Electronic Library (AISeL), 2017. URL: <http://hdl.handle.net/10125/41402>.
- 56 Sirisilp Kongsilp, Hathaichanok Chawmungkrung, and Takashi Komuro. Pop-out Tetris: An implementation of a tablet FTVR game. In *15th International Joint Conference on Computer Science and Software Engineering, JCSSE 2018, Nakhonpathom, Thailand, July 11-13, 2018*, pages 1–5. IEEE, 2018. doi:10.1109/JCSSE.2018.8457377.
- 57 Tsvi Kopelowitz and Robert Krauthgamer. Color-distance oracles and snippets. In Roberto Grossi and Moshe Lewenstein, editors, *27th Annual Symposium on Combinatorial Pattern Matching, CPM 2016, June 27-29, 2016, Tel Aviv, Israel*, volume 54 of *LIPICs*, pages 24:1–24:10. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPICs.CPM.2016.24.
- 58 Tsvi Kopelowitz, Seth Pettie, and Ely Porat. Higher lower bounds from the 3SUM conjecture. In Robert Krauthgamer, editor, *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 1272–1287. SIAM, 2016. doi:10.1137/1.9781611974331.ch89.
- 59 Laurens R. Krol, Sarah-Christin Freytag, and Thorsten O. Zander. Meyendtris: a hands-free, multimodal tetris clone using eye tracking and passive BCI for intuitive neuroadaptive gaming. In Edward Lank, Alessandro Vinciarelli, Eve E. Hoggan, Sriram Subramanian, and Stephen A. Brewster, editors, *Proceedings of the 19th ACM International Conference on Multimodal Interaction, ICMI 2017, Glasgow, United Kingdom, November 13 - 17, 2017*, pages 433–437. ACM, 2017. doi:10.1145/3136755.3136805.
- 60 Ian J. Lewis and Sebastian L. Beswick. Generalisation over details: The unsuitability of supervised backpropagation networks for Tetris. *Adv. Artif. Neural Syst.*, 2015:157983:1–157983:8, 2015. doi:10.1155/2015/157983.
- 61 Weifeng Li, Marc-Antoine Nüssli, and Patrick Jermann. Gaze quality assisted automatic recognition of social contexts in collaborative Tetris. In Wen Gao, Chin-Hui Lee, Jie Yang, Xilin



- Chen, Maxine Eskénazi, and Zhengyou Zhang, editors, *Proceedings of the 12th International Conference on Multimodal Interfaces / 7. International Workshop on Machine Learning for Multimodal Interaction, ICMI-MLMI 2010, Beijing, China, November 8-12, 2010*, pages 8:1–8:8. ACM, 2010. doi:10.1145/1891903.1891914.
- 62 Weifeng Li, Marc-Antoine Nüssli, and Patrick Jermann. Exploring personal aspects using eye-tracking modality in Tetris-playing. In *IEEE 13th International Workshop on Multimedia Signal Processing (MMSP 2011), Hangzhou, China, October 17-19, 2011*, pages 1–4. IEEE, 2011. doi:10.1109/MMSP.2011.6093841.
- 63 John K. Lindstedt and Wayne D. Gray. Extreme expertise: Exploring expert behavior in Tetris. In Markus Knauff, Michael Pauen, Natalie Sebanz, and Ipke Wachsmuth, editors, *Proceedings of the 35th Annual Meeting of the Cognitive Science Society, CogSci 2013, Berlin, Germany, July 31 - August 3, 2013*. cognitivesciencesociety.org, 2013. URL: <https://mindmodeling.org/cogsci2013/papers/0183/index.html>.
- 64 John K. Lindstedt and Wayne D. Gray. The "cognitive speed-bump": How world champion Tetris players trade milliseconds for seconds. In Stephanie Denison, Michael Mack, Yang Xu, and Blair C. Armstrong, editors, *Proceedings of the 42th Annual Meeting of the Cognitive Science Society - Developing a Mind: Learning in Humans, Animals, and Machines, CogSci 2020, virtual, July 29 - August 1, 2020*. cognitivesciencesociety.org, 2020. URL: <https://cogsci.mindmodeling.org/2020/papers/0020/index.html>.
- 65 Diana Lora, Antonio A. Sánchez-Ruiz, and Pedro A. González-Calero. Difficulty adjustment in Tetris with time series. In David Camacho, Marco Antonio Gómez-Martín, and Pedro Antonio González-Calero, editors, *Proceedings of the 3rd Congreso de la Sociedad Española para las Ciencias del Videjuego, Barcelona, Spain, June 29, 2016*, volume 1682 of *CEUR Workshop Proceedings*, pages 89–100. CEUR-WS.org, 2016. URL: [http://ceur-ws.org/Vol-1682/CoSeCiVi16\\_paper\\_10.pdf](http://ceur-ws.org/Vol-1682/CoSeCiVi16_paper_10.pdf).
- 66 Diana Lora, Antonio A. Sánchez-Ruiz, Pedro A. González-Calero, and Marco Antonio Gómez-Martín. Dynamic difficulty adjustment in Tetris. In Zdravko Markov and Ingrid Russell, editors, *Proceedings of the Twenty-Ninth International Florida Artificial Intelligence Research Society Conference, FLAIRS 2016, Key Largo, Florida, USA, May 16-18, 2016*, pages 335–339. AAAI Press, 2016. URL: <http://www.aaai.org/ocs/index.php/FLAIRS/FLAIRS16/paper/view/12824>.
- 67 Danica Mast and Sanne de Vries. Cooperative Tetris: The influence of social exertion gaming on game experience and social presence. In Ronald Poppe, John-Jules Ch. Meyer, Remco C. Veltkamp, and Mehdi Dastani, editors, *Intelligent Technologies for Interactive Entertainment - 8th International Conference, INTETAIN 2016, Utrecht, The Netherlands, June 28-30, 2016, Revised Selected Papers*, volume 178 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 115–123. Springer, 2016. doi:10.1007/978-3-319-49616-0\_11.
- 68 Taishi Oikawa, Chu-Hsuan Hsueh, and Kokolo Ikeda. Improving human players' t-spin skills in Tetris with procedural problem generation. In Tristan Cazenave, H. Jaap van den Herik, Abdallah Saffidine, and I-Chen Wu, editors, *Advances in Computer Games - 16th International Conference, ACG 2019, Macao, China, August 11-13, 2019, Revised Selected Papers*, volume 12516 of *Lecture Notes in Computer Science*, pages 41–52. Springer, 2019. doi:10.1007/978-3-030-65883-0\_4.
- 69 Panagiotis Oikonomou, Antonios N. Dadaliaris, Thanasis Loukopoulos, Athanasios Kakarountas, and Georgios I. Stamoulis. A Tetris-based legalization heuristic for standard cell placement with obstacles. In *7th International Conference on Modern Circuits and Systems Technologies, MOCAS 2018, Thessaloniki, Greece, May 7-9, 2018*, pages 1–4. IEEE, 2018. doi:10.1109/MOCAS.2018.8376559.
- 70 Zhan-He Ou and Ling-Hwei Chen. Hiding data in Tetris. In *International Conference on Machine Learning and Cybernetics, ICMLC 2011, Guilin, China, July 10-13, 2011, Proceedings*, pages 61–67. IEEE, 2011. doi:10.1109/ICMLC.2011.6016684.

## 13:18 How Fast Can We Play Tetris Greedily with Rectangular Pieces?

- 71 Zhan-He Ou and Ling-Hwei Chen. A steganographic method based on tetris games. *Inf. Sci.*, 276:343–353, 2014. doi:10.1016/j.ins.2013.12.024.
- 72 Jing Pang, Manuel Pravin, and Robert Tanihaha. Microblaze soft core based FPGA embedded system design of Tetris game. In Hamid R. Arabnia and Ashu M. G. Solo, editors, *Proceedings of the 2010 International Conference on Embedded Systems & Applications, ESA 2010, July 12-15, 2010, Las Vegas Nevada, USA*, pages 79–85. CSREA Press, 2010.
- 73 Nick Parlante. Nifty assignments: tetris on the brain. *ACM SIGCSE Bull.*, 33(4):25–27, 2001. doi:10.1145/572139.572162.
- 74 Georgios Patsis, Hichem Sahli, Werner Verhelst, and Olga De Troyer. Evaluation of attention levels in a Tetris game using a brain computer interface. In Sandra Carberry, Stephan Weibelzahl, Alessandro Micarelli, and Giovanni Semeraro, editors, *User Modeling, Adaptation, and Personalization - 21th International Conference, UMAP 2013, Rome, Italy, June 10-14, 2013, Proceedings*, volume 7899 of *Lecture Notes in Computer Science*, pages 127–138. Springer, 2013. doi:10.1007/978-3-642-38844-6\_11.
- 75 Colin Pinnell. Computer games for learning: An evidence-based approach (author: Richard e. mayer). *J. Educ. Technol. Soc.*, 18(4):523–524, 2015. URL: [https://www.j-ets.net/ETS/journals/18\\_4/40.pdf](https://www.j-ets.net/ETS/journals/18_4/40.pdf).
- 76 Gabriel Pires, Mario Torres, Nuno Casaleiro, Urbano Nunes, and Miguel Castelo-Branco. Playing Tetris with non-invasive BCI. In *2011 IEEE 1st International Conference on Serious Games and Applications for Health, SeGAH 2011, Braga, Portugal, November 16-18, 2011*, pages 1–6. IEEE Computer Society, 2011. doi:10.1109/SeGAH.2011.6165454.
- 77 Maximilian Probst. On the complexity of the (approximate) nearest colored node problem. In Yossi Azar, Hannah Bast, and Grzegorz Herman, editors, *26th Annual European Symposium on Algorithms, ESA 2018, August 20-22, 2018, Helsinki, Finland*, volume 112 of *LIPICs*, pages 68:1–68:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPICs.ESA.2018.68.
- 78 Mihai Pătraşcu. Towards polynomial lower bounds for dynamic problems. In *Proceedings of the Forty-Second ACM Symposium on Theory of Computing, STOC '10*, pages 603–610, New York, NY, USA, 2010. Association for Computing Machinery. doi:10.1145/1806689.1806772.
- 79 Jana Rambusch, Tarja Susi, Stefan Ekman, and Ulf Wilhelmsson. A literary excursion into the hidden (fan) fictional worlds of Tetris, starcraft, and dreamfall. In Tanya Krzywinska, Helen W. Kennedy, and Barry Atkins, editors, *Proceedings of the 2009 DiGRA International Conference: Breaking New Ground: Innovation in Games, Play, Practice and Theory, DiGRA 2009, London, UK, September 1-4, 2009*. Digital Games Research Association, 2009. URL: <http://www.digra.org/digital-library/publications/a-literary-excursion-into-the-hidden-fan-fictional-worlds-of-tetris-starcraft-and-dreamfall/>.
- 80 Catherine Sibert and Wayne D. Gray. When experts err: Using Tetris models to detect true errors from deliberate sub-optimal choices. In Ashok K. Goel, Colleen M. Seifert, and Christian Freksa, editors, *Proceedings of the 41th Annual Meeting of the Cognitive Science Society, CogSci 2019: Creativity + Cognition + Computation, Montreal, Canada, July 24-27, 2019*, page 3572. cognitivesciencesociety.org, 2019. URL: <https://mindmodeling.org/cogsci2019/papers/0880/index.html>.
- 81 Catherine Sibert, Wayne D. Gray, and John K. Lindstedt. Tetris-: Exploring human performance via cross entropy reinforcement learning models. In David C. Noelle, Rick Dale, Anne S. Warlaumont, Jeff Yoshimi, Teenie Matlock, Carolyn D. Jennings, and Paul P. Maglio, editors, *Proceedings of the 37th Annual Meeting of the Cognitive Science Society, CogSci 2015, Pasadena, California, USA, July 22-25, 2015*. cognitivesciencesociety.org, 2015. URL: <https://mindmodeling.org/cogsci2015/papers/0377/index.html>.
- 82 Catherine Sibert, John K. Lindstedt, and Wayne D. Gray. Tetris: Exploring human strategies via cross entropy reinforcement learning models. In Paul Bello, Marcello Guarini, Marjorie McShane, and Brian Scassellati, editors, *Proceedings of the 36th Annual Meeting of the Cognitive*

- Science Society, CogSci 2014, Quebec City, Canada, July 23-26, 2014*. cognitivesciencesociety.org, 2014. URL: <https://mindmodeling.org/cogsci2014/papers/743/>.
- 83 Özgür Simsek, Simón Algorta, and Amit Kothiyal. Why most decisions are easy in Tetris - and perhaps in other sequential decision problems, as well. In Maria-Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of the 33rd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 1757–1765. JMLR.org, 2016. URL: <http://proceedings.mlr.press/v48/simsek16.html>.
  - 84 Katta Spiel, Sven Bertel, and Fares Kayali. "Not another Z piece!": Adaptive difficulty in TETRIS. In Gloria Mark, Susan R. Fussell, Cliff Lampe, M. C. Schraefel, Juan Pablo Hourcade, Caroline Appert, and Daniel Wigdor, editors, *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, Denver, CO, USA, pages 5126–5131*. ACM, 2017. doi:10.1145/3025453.3025721.
  - 85 Katta Spiel, Sven Bertel, and Fares Kayali. Adapting gameplay to eye movements - an exploration with TETRIS. In Joan Arnedo, Lennart E. Nacke, Vero Vanden Abeele, and Z O. Toups, editors, *Extended Abstracts of the Annual Symposium on Computer-Human Interaction in Play Companion Extended Abstracts, Barcelona, Spain, October 22-25, 2019*, pages 687–695. ACM, 2019. doi:10.1145/3341215.3356267.
  - 86 Guo-Dong Su, Chin-Chen Chang, Chia-Chen Lin, and Zhi-Qiang Yao. Secure high capacity tetris-based scheme for data hiding. *IET Image Process.*, 14(17):4633–4645, 2020. doi:10.1049/iet-ipr.2019.1694.
  - 87 Guo-Dong Su, Chin-Chen Chang, Chia-Chen Lin, and Zhi-Qiang Yao. Erratum: Secure high capacity tetris-based scheme for data hiding. *IET Image Process.*, 15(12):3020, 2021. doi:10.1049/ipr2.12291.
  - 88 Tadao Takaoka. The reverse problem of range query. *Electron. Notes Theor. Comput. Sci.*, 78:281–292, 2003. doi:10.1016/S1571-0661(04)81018-1.
  - 89 Oscar Temprano. Complexity of a Tetris variant. *CoRR*, abs/1506.07204, 2015. arXiv:1506.07204.
  - 90 Jan van den Brand, Danupon Nanongkai, and Thatchaphol Saranurak. Dynamic matrix inverse: Improved algorithms and matching conditional lower bounds. In David Zuckerman, editor, *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2019, Baltimore, Maryland, USA, November 9-12, 2019*, pages 456–480. IEEE Computer Society, 2019. doi:10.1109/FOCS.2019.00036.
  - 91 Vladislav Daniel Veksler and Wayne D. Gray. State definition in the Tetris task: Designing a hybrid model of cognition. In *Proceedings of the International Conference on Cognitive Modeling, ICCM 2004, Pittsburgh, Pennsylvania, USA, July 30 - August 1, 2004*, pages 394–395, 2004. URL: <http://www.lrdc.pitt.edu/schunn/ICCM2004/proceedings/abstracts/Vekser.pdf>.
  - 92 Jianping Wang, Jun Chen, and Xiao-Min Li. Design a mobile Tetris game based on J2ME platform. In Chunfeng Liu, Jincai Chang, and Aimin Yang, editors, *Information Computing and Applications - Second International Conference, ICICA 2011, Qinhuangdao, China, October 28-31, 2011. Proceedings, Part II*, volume 244 of *Communications in Computer and Information Science*, pages 387–393. Springer, 2011. doi:10.1007/978-3-642-27452-7\_53.
  - 93 R. Ryan Williams. Faster all-pairs shortest paths via circuit complexity. *SIAM J. Comput.*, 47(5):1965–1985, 2018. doi:10.1137/15M1024524.
  - 94 Virginia Vassilevska Williams. On some fine-grained questions in algorithms and complexity. In *Proceedings of the International Congress of Mathematicians (ICM 2018)*, pages 3447–3487. WORLD SCIENTIFIC, 2019. doi:10.1142/9789813272880\_0188.
  - 95 Virginia Vassilevska Williams and Yinzhan Xu. Monochromatic triangles, triangle listing and APSP. In Sandy Irani, editor, *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 786–797. IEEE, 2020. doi:10.1109/FOCS46700.2020.00078.
  - 96 Jiajun Xu and Sam Huang. Tetris. *CoRR*, abs/1811.08614, 2018. arXiv:1811.08614.