

Yin-Yang Puzzles are NP-complete

Erik D. Demaine*

Jayson Lynch†

Mikhail Rudoy‡

Yushi Uno§

Abstract

We prove NP-completeness of Yin-Yang / Shiromaru-Kuromaru pencil-and-paper puzzles. Viewed as a graph partitioning problem, we prove NP-completeness of partitioning a rectangular grid graph into two induced trees (normal Yin-Yang), or into two induced connected subgraphs (Yin-Yang without 2×2 rule), subject to some vertices being pre-assigned to a specific tree/subgraph.

1 Introduction

The Yin-Yang puzzle is an over-25-year-old type of pencil-and-paper logic puzzle, the genre that includes Sudoku and many other puzzles made famous by e.g. Japanese publisher Nikoli.¹ Figure 1 shows a simple example of a puzzle and its solution. In general, a Yin-Yang puzzle consists of a rectangular $m \times n$ grid of unit squares, called *cells*, where each cell either has a black circle, has a white circle, or is empty. The goal of the puzzle is to fill each empty cell with either a black circle or a white circle to satisfy the following two constraints:

- **Connectivity constraint:** For each color (black and white), the circles of that color form a single connected group of cells, where connectivity is according to four-way orthogonal adjacency.
- **2×2 constraint:** No 2×2 square contains four circles of the same color.

In this paper, we prove NP-completeness of deciding whether a Yin-Yang puzzle has a solution, with or without the 2×2 constraint.

1.1 Graph Partitioning

We can view Yin-Yang puzzles as a type of graph partitioning problem, by taking the dual graph with a vertex for each unit-square cell and edges between orthogonally adjacent vertices/cells. The result is a rectangular $m \times n$ grid graph, with some vertices precolored black or white. The goal is to complete a black/white coloring of the vertices subject to dual versions of the constraints. The

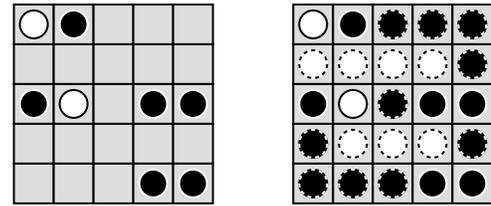


Figure 1: A simple Yin-Yang puzzle (left) and its unique solution (right). Circles added in the solution have dotted boundaries.

connectivity constraint above constrains each color class to induce a connected subgraph, so with this constraint alone, the problem is to partition the graph’s vertices into two connected induced subgraphs. We thus also prove NP-completeness of this graph partitioning problem:

Grid Graph Connected Partition Completion: Given an $m \times n$ grid graph $G = (V, E)$ and given a partition of V into A , B , and U , is there a partition of V into A' and B' such that $A \subseteq A'$, $B \subseteq B'$, and $G[A']$ and $G[B']$ are connected?

The 2×2 constraint forbids induced 4-cycles in each color class, which is the *thin* constraint of [7, 14]. Any larger-than-4 induced cycle in a color class must enclose a vertex of the opposite color, so by the connectivity constraint, only one color class can have such a cycle and it can have only one such cycle, which (if it exists) must be exactly the boundary vertices of the $m \times n$ grid graph. If we exclude the possibility of a single outer cycle (e.g., by restricting to instances that precolor at least one boundary vertex of each color), then the problem (with both constraints) is to partition the vertices of a rectangular $m \times n$ grid graph into two induced (connected) trees. We also prove NP-completeness of this graph partitioning problem:

Grid Graph Tree Partition Completion: Given an $m \times n$ grid graph $G = (V, E)$ and given a partition of V into A , B , and U , is there a partition of V into A' and B' such that $A \subseteq A'$, $B \subseteq B'$, and $G[A']$ and $G[B']$ are trees?

*Massachusetts Institute of Technology, USA, edemaine@mit.edu

†University of Waterloo, Canada, jayson.lynch@uwaterloo.ca

‡LeapYear Technologies, mrudoy@gmail.com

§Osaka Prefecture University, uno@cs.osakafu-u.ac.jp

¹However, Yin-Yang is not a Nikoli puzzle.

1.2 History

The origin of Yin-Yang is not clear. An early example of this puzzle is from 1994 in the (discontinued) Japanese puzzle magazine *Puzzler* [11] (or its original form in 1993). This reference calls the puzzle by the domestic Japanese name “白丸黒丸” (“Shiromaru-Kuromaru”, which means “white circle / black circle”). It seems that the puzzle or slight variations have been invented independently many times under different names. Most recently, it is often referred to as the “Yin-Yang” puzzle, and we could find the puzzle introduced in some puzzle books, magazines, and websites, e.g., [35].

The computational complexity of puzzles has seen significant study, partly for the recreational element but also because many puzzles have direct connection to important problems such as geometric packing/partitioning or path/tree drawing under constraints. Surveys have been written on the topic of the computational complexity of games and puzzles [12, 20, 29]. Many pencil-and-paper puzzles have been shown to be NP-complete, including: Bag / Corral [17], Country Road [24], Dosun-Fuwari [27], Fillomino [40], Fillmat [39] Hashiwokakero [6], Heyawake [23], Hiroimono / Goishi Hiroi [5], Hitori [20, Section 9.2], Jusoan [28] Kakuro / Cross Sum [41], Kurodoko [30], Kurotto [28], Light Up / Akari [33], LITS [34], Masyu / Pearl [18], Nonogram / Paint By Numbers [38], Numberlink [31], Nurikabe [32, 22], Pencils [36], Shakashaka [13, 2], Slitherlink [41, 40, 1], Spiral Galaxies / Tentai Show [19], Sto-Stone [4], Sudoku [41, 40], Tatamibari [3], Usowan [26], Yajilin [24], and Yosenabe [25].

Graph partitioning problems involve splitting the vertices of a graph into subsets based on various criteria. Common criteria include connectedness of the subgraphs, balancing the number or weight of vertices or edges in the partitions, and minimizing or maximizing the edge cut between partitions. Different objectives have a variety of applications; see [10] for a survey on the topic. Connected balanced partitions in grid graphs has been of specific interest with several NP-hardness results known, including 2-color weighted and only 3-rows [8], 3-color unweighted [9], and k -color in solid grid graphs [16]. On the positive side, Hettle et al. [21] give polynomial-time approximation algorithms for partitioning grid and planar graphs, and apply these to real-world police and fire-department districting problems. We use only two colors in a solid unweighted grid, but introduce the new constraint of pre-assigning more than one vertex to each partition. When exactly one vertex of each color is already assigned, the problem is often called a rooted partition problem.

2 Hardness Proof

All of the problems we consider are obviously in NP, by using the partition/solution as a witness. We show that solving $n \times n$ Yin-Yang puzzles is NP-hard with or without the 2×2 constraint by reductions from the following NP-hard problem [15]:

Planar 4-Regular Tree-Residue Vertex Breaking (TRVB): Given a planar 4-regular multigraph, is there a subset of vertices that, after being “broken”, results in a single connected tree? **Breaking** a vertex involves deleting that vertex from the graph and adding a degree-1 vertex to each of its neighbors (thus modifying the edges incident to the broken vertex to instead be incident to a newly created degree-1 vertex):



Figure 2 shows an example of this problem.

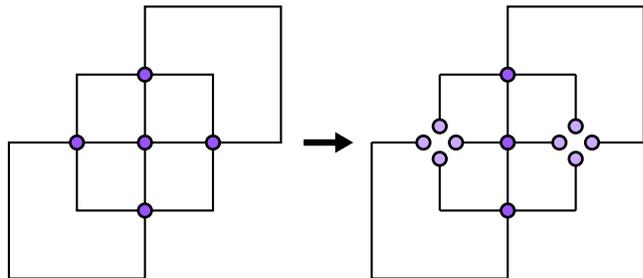


Figure 2: A 5-vertex instance of Planar 4-Regular TRVB (left) and a solution (right).

We will use the fact that planar maximum-degree-4 multigraphs can be drawn (with grid-routed edges) in a square grid of area $O(n^2)$ in polynomial time, as in Figure 2. We can use such an embedding for simple graphs (e.g., [37]), and adapt it to multigraphs by subdividing multiple edges and loops, applying the simple embedding, and removing the subdivision vertices.

Before diving into the reductions, we develop a tool that helps force local solutions to Yin-Yang puzzles:

Lemma 1 *In a valid Yin-Yang puzzle solution (with or without the 2×2 constraint), no 2×2 square can contain diagonally opposite white and black circles.*

Proof. Consider the black circles. These two circles are not orthogonally adjacent, and thus must be connected by a path of black circles. However, this path

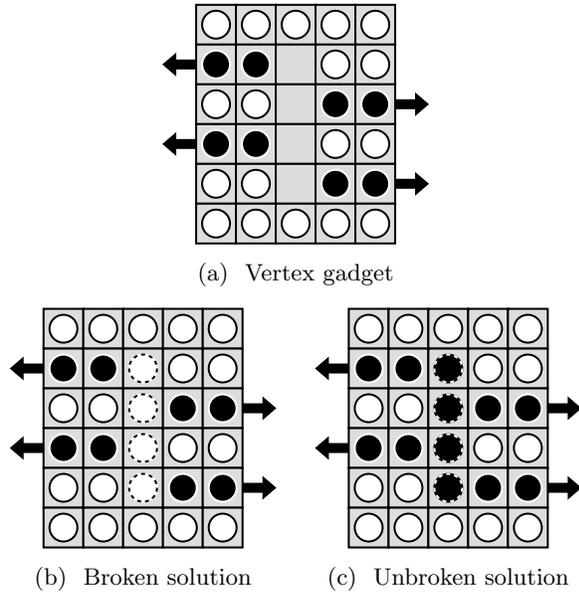


Figure 3: Vertex gadget without the 2×2 constraint and its two possible local solutions. The arrows represent the four outgoing edges.

will separate the pair of white circles which are also not orthogonally adjacent and thus must be connected by their own path of white circles. \square

2.1 Connected Partition: Without 2×2 Constraint

In this section, we prove NP-hardness of Grid Graph Connected Partition Completion, or equivalently, Yin-Yang puzzles without the 2×2 constraint. We present this proof first as it is simpler but uses the same ideas.

In this reduction, edges will be represented by orthogonal paths (*wires*) of black circles, and everything that is not an edge or vertex gadget will be filled with white circles.

Vertex Gadget. Figure 3a shows the vertex gadget. If a solution fills the top empty cell (say) black or white, then by repeated application of Lemma 1, all four empty cells must be filled the same color. In the local solution of Figure 3b, the black-circle wires are disconnected from each other; while in the local solution of Figure 3c, the black-circle wires are all connected together. Thus these two local solutions correspond to breaking and not breaking a vertex of the TRVB instance.

Layout. We take an orthogonal grid drawing of the given TRVB graph, and then scale the grid by a factor of 9. This scaling allows us to place the 9×9 tiled versions of the vertex gadget and edge gadgets (corners and straights) shown in Figure 4, which make it easy to align black-circle wires in row 3 and column 6. All re-

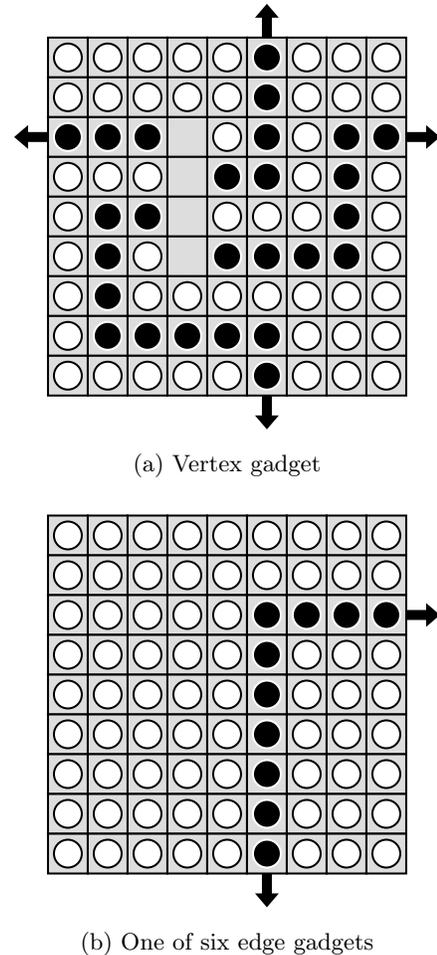


Figure 4: 9×9 tiling versions of gadgets without the 2×2 constraint.

maining cells are filled with white circles, leaving empty only the four cells in each vertex gadget.

Theorem 2 *It is NP-complete to decide whether there is a solution to a Yin-Yang puzzle without the 2×2 constraint, or Grid Graph Connected Partition Completion, on an $n \times n$ grid.*

Proof. If the TRVB instance has a solution, we fill broken vertices with four white circles (as in Figure 3b) and unbroken vertices with four black circles (as in Figure 3c). All edge gadgets touching an unbroken vertex are connected, and thus the fact that the TRVB solution is connected ensures the black circles form a single connected component. For a broken vertex, the added white circles connect the white circles in the four faces of the graph drawing. The TRVB solution being a tree ensures that this yields a single white connected component.

Conversely, consider any solution to the Yin-Yang puzzle. As argued above, each vertex gadget must be

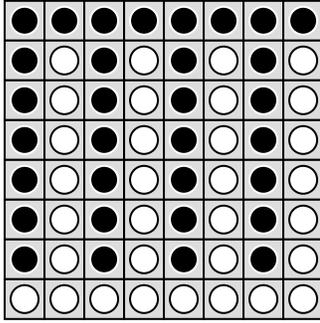


Figure 5: Background filler satisfying the 2×2 constraint.

solved using one of the two valid local solutions from Figure 3, and therefore we can translate this Yin-Yang solution to an assignment of whether to break each vertex in the TRVB instance. We claim that this assignment is in fact a solution to the TRVB instance. The region containing the black circles in the Yin-Yang solution has the same shape (in particular, topology) as the graph resulting from breaking the broken vertices in the candidate solution to the TRVB instance. If the resulting graph were disconnected, then the black circles would also form a disconnected region, contradicting the Yin-Yang connectivity constraint on black. If the resulting graph had a cycle, then the black circles would also form a nontrivial cycle, which would separate the white circles interior and exterior to that cycle, contradicting the Yin-Yang connectivity constraint on white. Therefore the resulting graph is connected and acyclic, so we have a solution to the TRVB instance. \square

2.2 Tree Partition: With 2×2 Constraint

In this section, we prove NP-hardness of Grid Graph Tree Partition Completion as well as Yin-Yang puzzles (with both constraints). We fully precolor the boundary vertices to not form a cycle, so these problems become equivalent. The reduction idea is the same as the proof without the 2×2 constraint from Section 2.1, but respecting this constraint requires a more complicated filler, and because of this filler, a more complex vertex gadget. To get a sense of the overall structure, refer ahead to Figure 10 for a full example.

2.2.1 Background Filler

The background filler consists of alternating columns of white and black circles, except for a full row of black circles at the top and a full row of white circles at the bottom (to ensure connectivity); see Figure 5. On top of this filler gadget, we draw vertex and edge gadgets.

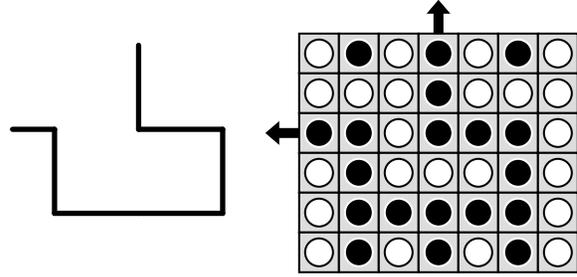


Figure 6: An edge route (left) and the corresponding edge gadget (right).

2.2.2 Edge Gadget

We again represent edges as orthogonal paths (wires) of black circles but now, wherever an edge travels horizontally, we also add a row of white circles immediately above the path, except where the edge turns upward; see Figure 6. Because of the background filler, each horizontal segment of an edge gadget will have black downward tendrils (paths) from every other column, similar to the downward black tendrils from the top row of the filler. (Similarly, the white circles immediately above a horizontal segment of an edge gadget will have white upward tendrils from every other column, similar to the upward white tendrils from the bottom row of the filler.) Edge gadgets can travel vertically only on the black parity of the background filler, so that these circles of the edge gadget match the background filler.

2.2.3 Vertex Gadget

Figure 7 shows the vertex gadget. By the 2×2 constraint, the leftmost two empty cells must be filled with circles of opposite color; refer to Figure 8. Once this choice gets made, repeated application of Lemma 1 forces the coloring of the horizontal rows of empty squares to be uniform and opposite from each other, and then forces the rightmost two empty cells to match the leftmost two empty cells. The top three empty cells are then forced to be filled in with circles of the same color as the bottom row, in order to prevent local isolation of black or white circles.

Figure 8 shows the two possible resulting local solutions. Figure 8a corresponds to breaking the TRVB vertex, as it keeps the four incident edges disconnected from each other; while Figure 8b corresponds to not breaking the corresponding TRVB vertex, as it connects together the four incident edges. Both solutions also connect together the five white paths at the top and the two outermost white paths on the bottom, and the broken solution further connects these white paths to the three middle white paths at the bottom. These connections correspond to exactly one white region per face around the vertex: one face for a broken vertex and

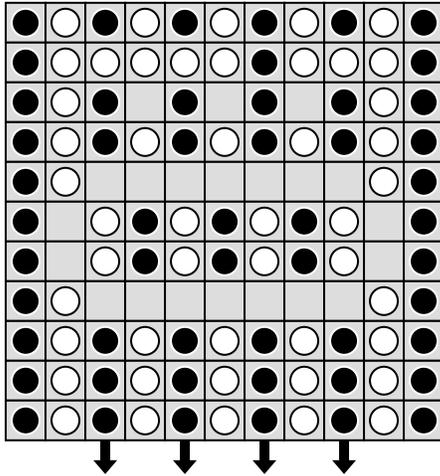


Figure 7: Vertex gadget. The arrows represent the four outgoing edges.

four faces for an unbroken vertex.

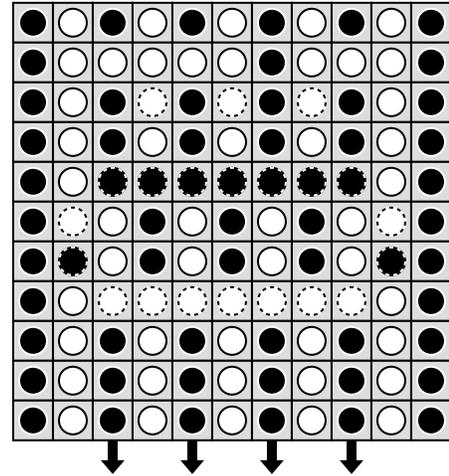
2.2.4 Reduction Layout

We take an orthogonal grid drawing of the given TRVB graph, and then scale the grid by a factor of 16. This scaling allows us to place the 16×16 tiled versions of the vertex gadget and edge gadgets (corners and straights) shown in Figure 9, which make it easy to align black-circle wires in row 12 and column 11. Any absent 16×16 squares get filled with alternating columns of black and white. Then we add an extra row at the top and bottom of the puzzle, filled with black and white circles respectively, to complete the filler of Figure 5. Finally, for one topmost horizontal segment of an edge gadget, in a black-parity column, we change one of the cells above the segment from white to black, thereby attaching the edge gadget to a black tendril and thus the top row of black circles; we call the changed cell the *exceptional cell*.

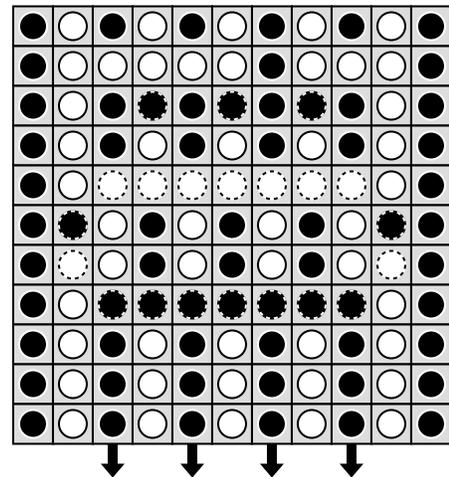
Figure 10 shows a complete example of the reduction applied to the instance from Figure 2, and Figure 11 shows the corresponding solution.

Theorem 3 *It is NP-complete to decide whether there is a solution to a Yin-Yang puzzle (with both constraints), or Grid Graph Tree Partition Completion, on an $n \times n$ grid.*

Proof. Because our reduction instance has black and white circles on the boundary, the Yin-Yang puzzle with 2×2 constraint is equivalent to Grid Graph Tree Partition Completion. Furthermore, if the black circles form a connected and acyclic subset of cells, then so do the white circles: if the white circles formed an induced cycle of length > 4 , then there would be black circles both interior (because the cycle is induced) and exterior (on



(a) Broken vertex solution.



(b) Unbroken vertex solution.

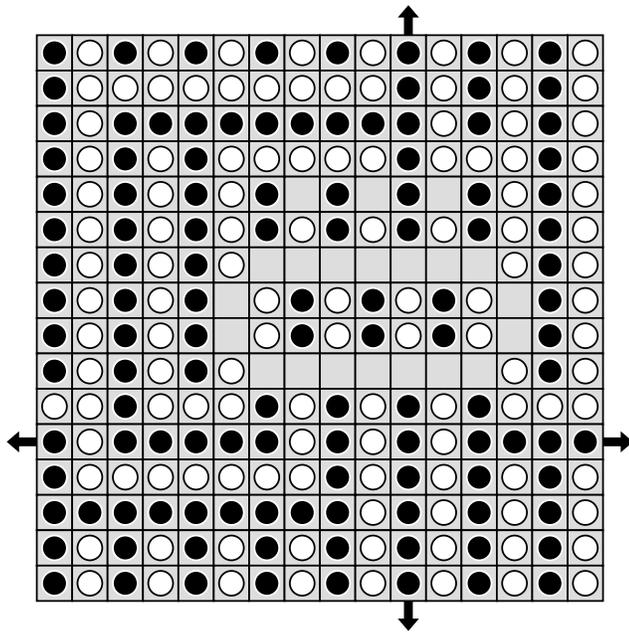
Figure 8: The two local solutions to the vertex gadget of Figure 7.

the boundary), a contradiction. Therefore, it suffices to prove that the black circles are connected and acyclic if and only if the chosen broken/unbroken solutions from Figure 8 for each vertex gadget corresponds to a solution to the TRVB instance.

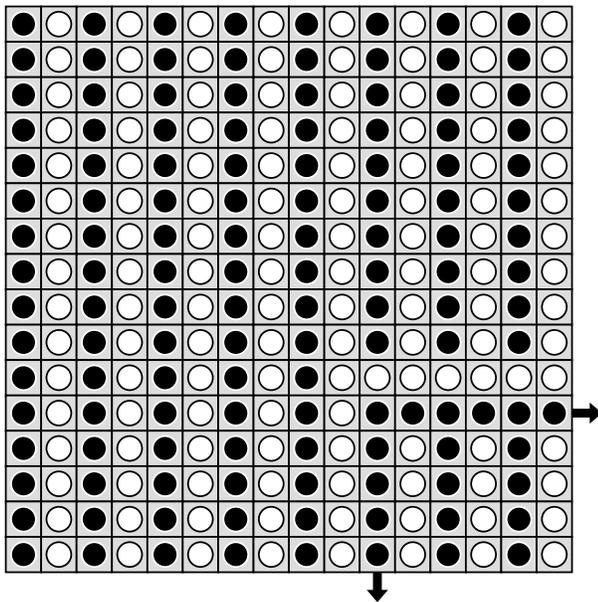
Define the *important* cells to consist of the following:

1. black edge wires, or more precisely, the shortest paths among black circles connecting pairs of the vertex gadgets' ports (marked with arrows in Figure 7); and
2. the bottom row of initially empty cells in each vertex gadget (instance of Figure 7), together with the shortest paths of black circles within the gadget connecting those cells to the ports (arrows).

The important cells directly represent the connectivity of the TRVB instance, so the important black circles



(a) Vertex gadget



(b) One of six edge gadgets

Figure 9: 16×16 tiling versions of the vertex and edge gadgets from Figures 7 and 6.

are connected and acyclic if and only if the Yin-Yang solution corresponds to a TRVB solution.

Thus it suffices to show that the important black circles are connected and acyclic if and only if all of the black circles are connected and acyclic. Unimportant black circles are either part of a vertex gadget, part of the top row, or part of a downward black tendril from the top row or a horizontal segment of an edge gadget.

Figure 8 shows that unimportant black circles within a vertex gadget are all attached to the downward black tendrils above the gadget, with one acyclic connected component per tendril, aside from a few black circles (in the unbroken case) attached directly to important black circles. For unimportant black circles in black tendrils, we can trace the tendrils up to find their connection to either important black circles or the top row. (Tendrils are never connected to black circles at their bottom, except to $O(1)$ black circles within a vertex gadget.) The black tendrils that drop from an important black circle are all connected together if and only if the important black circles are themselves connected. The black tendrils that drop from the top row are all connected together via the top row of unimportant black circles, and the exceptional cell connects one of these tendrils to the important black circles. Therefore the unimportant black circles form trees attached to the important black cells, so they do not affect connectivity or acyclicity. \square

3 Open Problems

Our reduction from Planar 4-regular TRVB is parsimonious (preserves the number of solutions). It is unknown whether the Another Solution Problem or counting versions of TRVB are (NP- or #P-) hard, but such results would carry over to Yin-Yang puzzles.

In our reductions, we fill almost the entire board with circles. However, elegant puzzles tend to have only a few pre-marked circles and so one may wonder whether this problem remains hard with a much sparser clue set. It seems likely small modifications to our reduction would allow for reductions with only $O(n)$ circles, but getting $o(n)$ would require fundamentally different techniques and some sort of clever encoding of problems in the relative spacing of the circles. On the extreme, it seems reasonably likely this problem is FPT with respect to the number of pre-assigned circles.

Both from a computational and puzzle design standpoint, we could imagine generalizing the number of colors of circles to more than two, or looking at other families of graphs. The 2×2 constraint can generalize to forbidding all vertices around a face from being the same color in an embedded graph. It seems very natural to look at such puzzles on triangular and hexagonal grids and in those cases the three-fold symmetry may aesthetically adapt to three colors. Generalizing the number of colors trivially remains NP-hard, as we can simply add a row for each additional color at the top of our two color reduction, ensuring those new colors will not be able to be placed. We also believe similar proof techniques will work for hexagonal and triangular grids.

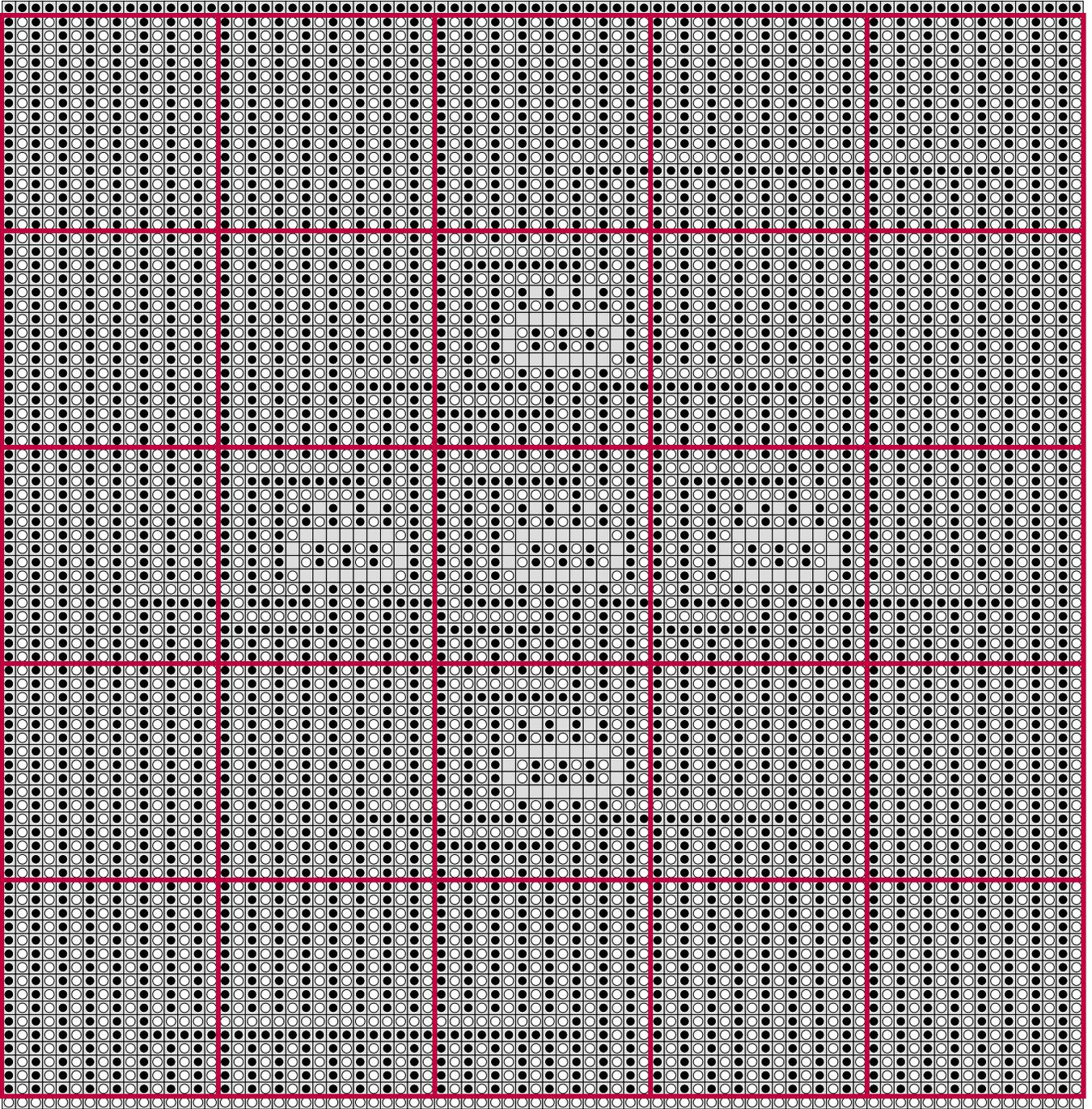


Figure 10: Full reduction from the 5-vertex instance of TRVB from Figure 2.

Acknowledgments

This research started at the 34th Bellairs Winter Workshop on Computational Geometry, co-organized by Erik Demaine and Godfried Toussaint in 2019. We thank the other workshop participants for fruitful discussions and for providing an inspiring research atmosphere.

Research supported in part by NSERC and JSPS KAKENHI Grant Numbers JP17K00017, JP21K11757

and JP20H05964.

Most figures of this paper were drawn using SVG Tiler [<https://github.com/edemaine/svgtiler>]. Source files for these figures are freely available [<https://github.com/edemaine/yin-yang-svgtiler>].

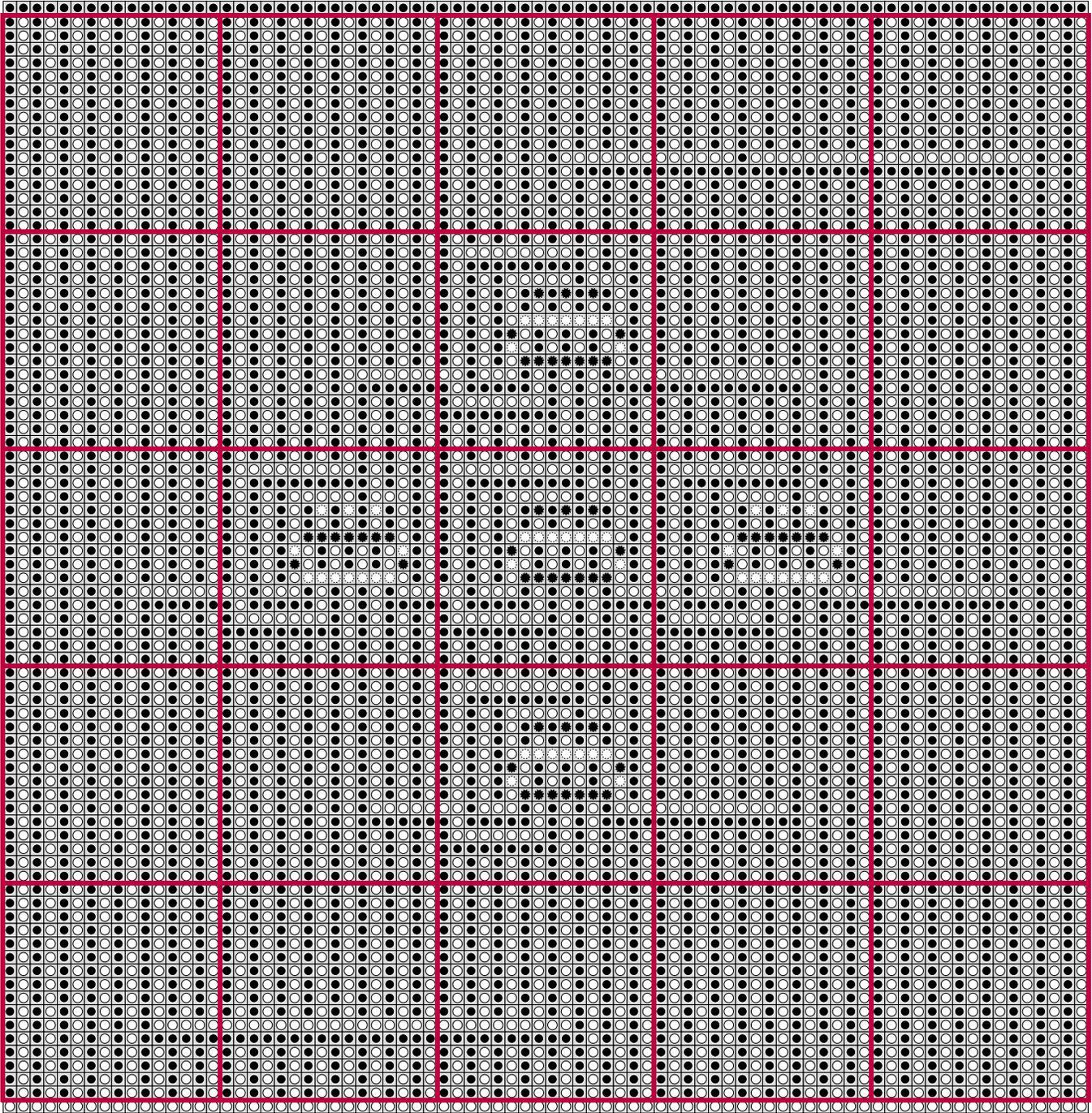


Figure 11: Solution to the puzzle in Figure 10 corresponding to the TRVB solution in Figure 2.

References

- [1] Zachary Abel, Jeffrey Bosboom, Erik D. Demaine, Linus Hamilton, Adam Hesterberg, Justin Kopinsky, Jayson Lynch, and Mikhail Rudoy. Who witnesses The Witness? Finding witnesses in The Witness is hard and sometimes impossible. In *Proceedings of the 9th International Conference on Fun with Algorithms (FUN 2018)*, pages 3:1–3:21, La Maddalena, Italy, June 2018.
- [2] Aviv Adler, Michael Biro, Erik Demaine, Mikhail Rudoy, and Christiane Schmidt. Computational complexity of numberless Shakashaka. In *Proceedings of the 27th Canadian Conference on Computational Geometry (CCCG 2015)*, Kingston, Canada, August 2015.
- [3] Aviv Adler, Jeffrey Bosboom, Erik D. Demaine,

- Martin L. Demaine, Quanquan C. Liu, and Jayson Lynch. Tatamibari is NP-complete. In *Proceedings of the 10th International Conference on Fun with Algorithms (FUN 2020)*, LIPIcs, 2020.
- [4] Addison Allen and Aaron Williams. Sto-stone is NP-complete. In *Proceedings of the 30th Canadian Conference on Computational Geometry (CCCG 2018)*, pages 28–34, 2018.
- [5] Daniel Andersson. HIROIMONO is NP-complete. In *Proceedings of the 4th International Conference on Fun with Algorithms (FUN 2007)*, volume 4475 of *Lecture Notes in Computer Science*, pages 30–39, 2007.
- [6] Daniel Andersson. Hashiwokakero is NP-complete. *Information Processing Letters*, 109(19):1145–1146, 2009.
- [7] Esther M. Arkin, Sándor P. Fekete, Kamrul Islam, Henk Meijer, Joseph S. B. Mitchell, Yurai Núñez-Rodríguez, Valentin Polishchuk, David Rappaport, and Henry Xiao. Not being (super) thin or solid is hard: A study of grid hamiltonicity. *Computational Geometry: Theory and Applications*, 42(6–7):582–605, 2009.
- [8] Ronald Becker, Isabella Lari, Mario Lucertini, and Bruno Simeone. Max-min partitioning of grid graphs into connected components. *Networks*, 32(2):115–125, 1998.
- [9] Cedric Berenger, Peter Niebert, and Kevin Perrot. Balanced connected partitioning of unweighted grid graphs. In *Proceedings of the 43rd International Symposium on Mathematical Foundations of Computer Science (MFCS 2018)*, volume 117 of *LIPIcs*, 2018.
- [10] Aydın Buluç, Henning Meyerhenke, Ilya Safro, Peter Sanders, and Christian Schulz. Recent advances in graph partitioning. In *Selected Results and Surveys on Algorithm Engineering*, volume 9220 of *Lecture Notes in Computer Science*, pages 117–158. Springer, 2016.
- [11] 世界文化社. しろまるくろまる. パズラー (*Puzler*), 150, May 1994.
- [12] Erik D. Demaine and Robert A. Hearn. Playing games with algorithms: Algorithmic combinatorial game theory. In *Games of No Chance 3*, pages 3–56. Cambridge University Press, 2009. arXiv:cc.CC/0106019.
- [13] Erik D. Demaine, Yoshio Okamoto, Ryuhei Uehara, and Yushi Uno. Computational complexity and an integer programming model of Shakashaka. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E97-A(6):1213–1219, 2014.
- [14] Erik D. Demaine and Mikhail Rudoy. Hamiltonicity is hard in thin or polygonal grid graphs, but easy in thin polygonal grid graphs. arXiv:1706.10046, 2017. <https://arXiv.org/abs/1706.10046>.
- [15] Erik D. Demaine and Mikhail Rudoy. Tree-Residue Vertex-Breaking: a new tool for proving hardness. In *Proceedings of the 16th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT 2018)*, volume 101 of *LIPIcs*, pages 32:1–32:14, 2018.
- [16] Andreas Emil Feldmann. Fast balanced partitioning is hard even on grids and trees. *Theoretical Computer Science*, 485:61–68, 2013.
- [17] Erich Friedman. Corral puzzles are NP-complete. <https://erich-friedman.github.io/papers/corral.pdf>, August 2002.
- [18] Erich Friedman. Pearl puzzles are NP-complete. <https://erich-friedman.github.io/papers/pearl.pdf>, August 2002.
- [19] Erich Friedman. Spiral Galaxies puzzles are NP-complete. <https://erich-friedman.github.io/papers/spiral.pdf>, March 2002.
- [20] Robert A. Hearn and Erik D. Demaine. *Games, Puzzles, and Computation*. A K Peters, July 2009.
- [21] Cyrus Hettle, Shixiang Zhu, Swati Gupta, and Yao Xie. Balanced districting on grid graphs with provable compactness and contiguity. arXiv:2102.05028, 2021. <https://arXiv.org/abs/2102.05028>.
- [22] Markus Holzer, Andreas Klein, and Martin Kutrib. On the NP-completeness of the NURIKABE pencil puzzle and variants thereof. In *Proceedings of the 3rd International Conference on Fun with Algorithms (FUN 2004)*, pages 77–89, Isola d’Elba, Italy, May 2004.
- [23] Markus Holzer and Oliver Ruepp. The troubles of interior design—a complexity analysis of the game Heyawake. In *Proceedings of the 4th International Conference on Fun with Algorithms (FUN 2007)*, volume 4475 of *Lecture Notes in Computer Science*, pages 198–212, 2007.
- [24] Ayaka Ishibashi, Yuichi Sato, and Shigeki Iwata. NP-completeness of two pencil puzzles: Yaajilin and Country Road. *Utilitas Mathematica*, 88:237–246, 2012.

- [25] Chuzo Iwamoto. Yosenabe is NP-complete. *Journal of Information Processing*, 22(1):40–43, 2014.
- [26] Chuzo Iwamoto and Masato Haruishi. Computational complexity of usowan puzzles. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 101(9):1537–1540, 2018.
- [27] Chuzo Iwamoto and Tatsuaki Ibusuki. Dosunfuwari is NP-complete. *Journal of Information Processing*, 26:358–361, 2018.
- [28] Chuzo Iwamoto and Tatsuaki Ibusuki. Polynomial-time reductions from 3SAT to kurotto and juosan puzzles. *IEICE Transactions on Information and Systems*, 103(3):500–505, 2020.
- [29] Graham Kendall, Andrew Parkes, and Kristian Spoerer. A survey of NP-complete puzzles. *ICGA Journal*, 31(1):13–34, 2008.
- [30] Jonas Kölker. Kurodoko is NP-complete. *Journal of Information Processing*, 20(3):694–706, 2012.
- [31] Kouichi Kotsuma and Yasuhiko Takenaga. NP-completeness and enumeration of Number Link puzzle. *IEICE Technical Report*, 109(465):1–7, March 2010.
- [32] Brandon McPhail. The complexity of puzzles. Undergraduate thesis, Reed College, Portland, Oregon, 2003.
- [33] Brandon McPhail. Light Up is NP-complete. <http://www.mountainvistasoft.com/docs/lightup-is-np-complete.pdf>, 2005.
- [34] Brandon McPhail. Metapuzzles: Reducing SAT to your favorite puzzle. CS Theory talk, December 2007.
- [35] Gareth Moore. *Paper, Pencil & You: Mindfulness: Relaxing Brain-Training Puzzles for Stressed-Out People*. Greenfinch, 2020.
- [36] Daniel Packer, Sophia White, and Aaron Williams. A paper on Pencils: A pencil and paper puzzle: Pencils is NP-complete, pencils. In *Proceedings of the 30th Canadian Conference on Computational Geometry (CCCG 2018)*, page 35, 2018.
- [37] Achilleas Papakostas and Ioannis G. Tollis. Algorithms for area-efficient orthogonal drawings. *Computational Geometry: Theory and Applications*, 9(1):83–110, 1998.
- [38] Nobuhisa Ueda and Tadaaki Nagao. NP-completeness results for NONOGRAM via parsimonious reductions. Technical Report TR96-0008, Department of Computer Science, Tokyo Institute of Technology, Tokyo, Japan, May 1996.
- [39] Akihiro Uejima and Hiroaki Suzuki. Fillmat is NP-complete and ASP-complete. *Journal of Information Processing*, 23(3):310–316, 2015.
- [40] Takayuki Yato. Complexity and completeness of finding another solution and its application to puzzles. Master’s thesis, University of Tokyo, Tokyo, Japan, January 2003.
- [41] Takayuki Yato and Takahiro Seta. Complexity and completeness of finding another solution and its application to puzzles. *IEICE Transactions on Fundamentals of Electronics, Communications, and Computer Sciences*, E86-A(5):1052–1060, 2003. Also IPSJ SIG Notes 2002-AL-87-2, 2002.