Hyperbolic Minesweeper Is in P

Eryk Kopczyński
Institute of Informatics, University of Warsaw, Poland
http://www.mimuw.edu.pl/~erykk/
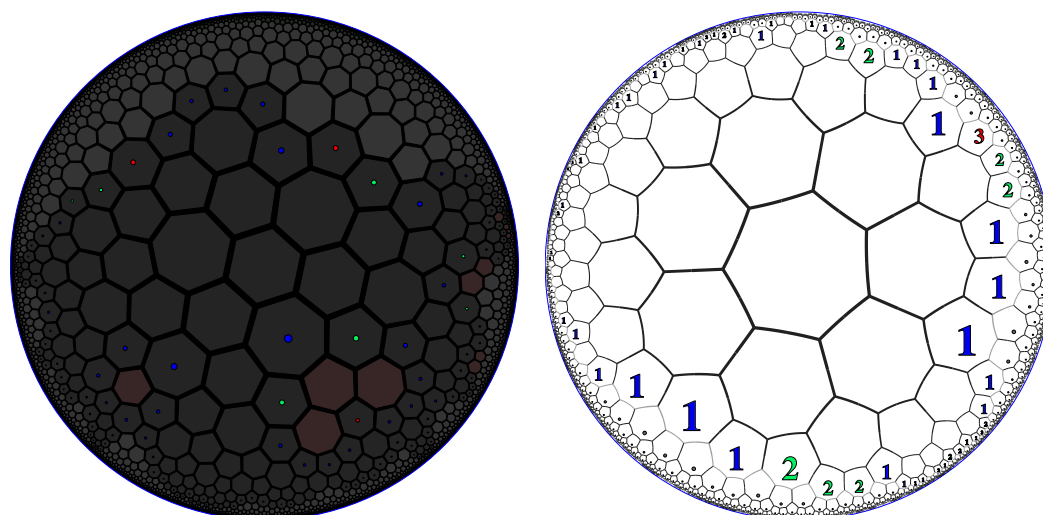erykk@mimuw.edu.pl

──── **Abstract** ────

We show that, while Minesweeper is NP-complete, its hyperbolic variant is in P. Our proof does not rely on the rules of Minesweeper, but is valid for any puzzle based on satisfying local constraints on a graph embedded in the hyperbolic plane.

1 Introduction

(a) graphical mode, bitruncated tessellation.

(b) text mode, regular tessellation.

■ **Figure 1** Hyperbolic Minesweeper [7]. In (a), the default settings are used (bitruncated order-3 heptagonal tessellation, numbers of adjacent mines are color-coded; some of the mines that the players is sure of are marked red). In (b) we play on an order-3 heptagonal tessellation, and numbers are shown.

Minesweeper is a popular game included with many computer systems; it also exists in the puzzle form. In the puzzle form, every cell in a square grid either contains a number or is empty. The goal of the puzzle is to assign mines to the empty squares in such a way that every number n is adjacent (orthogonally or diagonally) to exactly n mines.

This puzzle is a well-known example of a NP-complete problem [6]. Its popularity has also spawned many variants played on different grids, from changing the tessellation of the plane, to changing the number of dimensions (six-dimensional implementations of Minesweeper exist) to changing the underlying geometry.

In this paper we will be playing Minesweeper on a tessellation of the hyperbolic plane. Minefield, a land based on hyperbolic Minesweeper is available in HyperRogue [7]; this implementation differs from the standard Minesweeper in multiple ways, e.g., by being played on an infinite board, however, the basic idea is the same. Another implementation is Warped Mines for iOS [14], which has the same rules as the usual Minesweeper except the board, which is a bounded subset of the order-3 heptagonal tiling of the hyperbolic plane.

From the point of view of a computer scientist, the most important distinctive property of hyperbolic geometry is exponential growth: the area of a circle of radius r grows exponentially with r. It is also more difficult to understand than Euclidean geometry. While these properties often cause computational geometry problems to be more difficult, it also gives hyperbolic geometry applications in data visualization [8, 10] and data analysis [11].

In this paper we show that hyperbolic geometry makes Minesweeper easier: the hyperbolic variant of Minesweeper is in P. Our proof will not rely on the specific rules of Minesweeper nor work with any specific tessellation of the hyperbolic plane; instead, it will work with any puzzle based on satisfying local constraints, on any graph that naturally embeds into the hyperbolic plane.

2   Hyperbolic Geometry

We denote the hyperbolic plane with $\mathbb{H}^2$. Since this paper requires only the basic understanding of hyperbolic geometry, we will not include the formal definition; see [2], or [7] for an intuitive introduction. Figures 1 and 2 shows the hyperbolic plane in the Poincaré disk model, which is a projection which represents angles faithfully, but not the distances: the scale gets smaller and smaller as we get closer to the circle bounding the model. In particular, all the heptagons in Figure 1b are the same size, all the heptagons in Figure 1a are the same size, and all the hexagons in Figure 1a are the same size. We denote the distance between two points $x, y \in \mathbb{H}^2$ by $\delta(x, y)$. The set of points in distance at most r from x is denoted with $B(x, r)$. The area of $B(x, r)$ is $2\pi(\cosh r - 1)$, which we denote with area(r). The perimeter of $B(x, r)$ is $2\pi \sinh r$, which we denote with peri(r). Note that both area and peri grow exponentially: area$(r) = \Theta(e^r)$ and peri$(r) = \Theta(e^r)$.
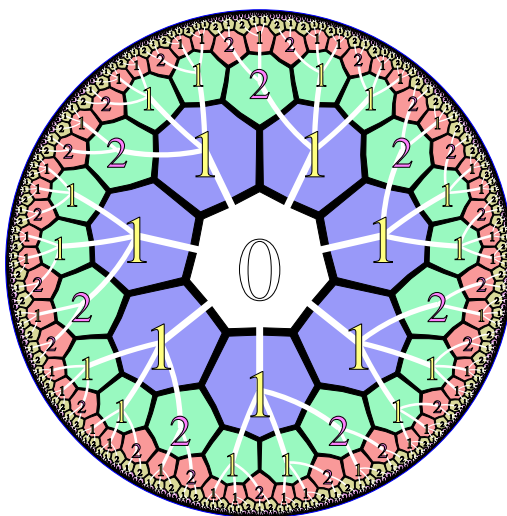
Figure 2 Exponential growth in the hyperbolic plane.

We include Figure 2 as a simple introduction to the structure of the hyperbolic plane and its exponential growth. This is the order-3 heptagonal tessellation [9]. The numbers correspond to the number of adjacent tiles which are closer to the center tile. This tessellation can be constructed as follows. We construct a tree of tiles: the central tile has seven children of type 1; each tile of tile 1 has two children of type 1, and one child of type 2; each tile of tile 2 has one child of type 1 and one child of type 2. To produce the actual tessellation, we also connect adjacent tiles on the same level, and to the leftmost child of the tile on the right. The hyperbolic plane can be seen as a continuous version of the graph obtained using this construction. Since every tile has at least two children, there are exponentially many tiles in r steps from the central tile.

3 Hyperbolic Graph

We need a suitable definition of a hyperbolic graph. The obvious definition, obtained by changing "plane" to "hyperbolic plane" in the definition of a planar graph, is equivalent to the definition of the planar graph (the plane and the hyperbolic plane are homeomorphic). Another definition found in literature is the notion of Gromov δ-hyperbolic graph [1]; this definition is based on the observation that, in the hyperbolic plane, triangles are thin, i.e., for any three vertices a, b, c, every point on any shortest path from a to c must be in distance at most δ from either the shortest path from a to b, or the shortest path from b to c. The parameter δ measures tree-likeness of the graph (it can be easily seen that trees are 0-hyperbolic). However, this definition is not suitable for us, because every graph G becomes 2-hyperbolic when we add a vertex v_* connected to every $v \in V(G)$; our main result is not true for such graphs. We propose the following definition:

▶ **Definition 1.** *A (r, d)-**hyperbolic graph** is a graph $G = (V, E)$ such that there exists an embedding $m : V \to \mathbb{H}^2$ such that:*
- *for $v_1 \neq v_2$, $\delta(m(v_1), m(v_2)) > r$,*
- *for $\{v_1, v_2\} \in E$, $\delta(m(v_1), m(v_2)) < d$,*
- *if we draw every edge $\{v_1, v_2\} \in E$ as a straight line segment between $m(v_1)$ and $m(v_2)$, these edges do not cross, nor they get closer to vertices other than v_1 or v_2 than $r/2$.*

Intuitively, $r/2$ is the radius of every vertex; this parameter bounds the density of our graph embedding. The parameter d gives the maximum distance between two vertices connected with an edge.

This definition includes finite subsets of all regular and semiregular hyperbolic tessellations (such as the ones shown in Figure 1). We denote $N(v)$ to be the neighborhood of $v \in V$, i.e., $\{v\} \cup \{w \in V : \{v, w\} \in E\}$. (Our proof also works with neighborhoods of larger radius.)

Remark. All (r, d)-hyperbolic graphs have degree bounded by a constant (for fixed r, d). Indeed, let $v \in V$. For every $w \in N(v)$, the circles $B(w, \frac{r}{2})$ are disjoint, and they fit in $B(v, d + \frac{r}{2})$. Therefore, $|N| \leq \text{area}(d + \frac{r}{2})/\text{area}(\frac{r}{2})$.

4 Hyperbolic Local Constraint Satisfaction Problem

Below we state our main result.

▶ **Theorem 2.** *Fix the set of colors K and the parameters (r, d). The following problem (Hyperbolic Local Constraint Satisfaction Problem, HLCSP) can be solved in polynomial time:*
INPUT:
- *a (r, d)-hyperbolic graph $G = (V, E)$;*
- *for every vertex $v \in V$, $m(v)$, a subset of $K^{N(v)}$.*

OUTPUT: *Is there a coloring $c : V \to K$ such that for every $v \in V$, $c_{|N(v)} \in m(v)$?*

HLCSP generalizes hyperbolic minesweeper. We have two colors (no mine and mine), and for every $v \in V$ containing a number k, the constraint $m(v)$ says that v contains no mine itself, and exactly k vertices in $N(v)$ contain a mine.

To prove Theorem 2, it is enough to prove that the following problem (HECSP) is in P.

▶ **Theorem 3.** *HLCSP reduces to the Hyperbolic Edge Constraint Satisfaction Problem (HECSP) given as follows:*
   INPUT:
   - *a (r, d)-hyperbolic graph $G = (V, E)$;*
   - *for every edge $e \in E$, $m(e)$, a subset of K^e.*

   OUTPUT: *Is there a coloring $c : V \to K$ such that for every $e \in E$, $c_{|e} \in m(e)$?*
   (Note: this reduction changes the set of admissible colors K.)

Proof. Let (G, m) be the instance of HLCSP. We will be coloring V using colors $K' = \{1, \ldots, k\}$, where k is the greatest number of elements of $m(v)$ (since (r, d)-hyperbolic graphs have bounded degree and K is fixed, k is also bounded). Enumerate every elements of $m(v)$ with one color from K'. For $e = \{v_1, v_2\} \in E$, $m(e)$ is the set of all colorings $c : \{v_1, v_2\} \to K$ which are consistent, i.e., k_1 denotes $c_1 \in m(v_1)$ and k_2 denotes $c_2 \in m(v_2)$, and c_1 equals c_2 on all the vertices in $N(v_1) \cap N(v_2)$.   ◄

5   Proof of Theorem 2

We will be using the following result [12, 3]:

▶ **Theorem 4.** *Given a planar graph $G = (V, E)$ and a number t, it is possible to either find a $t \times t$ grid as a minor of G, or produce a tree decomposition of G of width $\leq 5t - 6$, in time $O(n^2 \log(n))$, where $n = |V|$.*

▶ **Definition 5.** *A tree decomposition of width w of a graph $G = (V, E)$ is (V_T, E_T, X) where (V_T, E_T) is a tree, and $X : V_T \to P(V)$ is a mapping which assigns a subset of V of cardinality at most $w + 1$ to every vertex in V_T, such that:*
   - *For every $v \in V$, the set of vertices $b \in V_T$ such that $v \in X_b$ is connected,*
   - *For every $e \in E$, there exists a $b \in V_T$ such that $e \subseteq X_b$.*

We can assume that our tree is rooted in $r \in V_T$. For $b \in V_T$, let X_b^+ be the union of $X_{b'}$ for all b' which are descendants of b.

▶ **Lemma 6.** *Without loss of generality we can assume that every $b \in V_T$ falls into one of the following cases:*

   - *b is a leaf and $|X_b| = 1$,*
   - *b has a single child b' and $X_{b'} = X_b \cup \{v\}$,*
   - *b has a single child b' and $X_{b'} = X_b - \{v\}$,*
   - *b has two children b_1 and b_2, $X_b = X_{b_1} = X_{b_2}$, and $X_{b_1}^+ - X_b$ and $X_{b_2}^+ - X_b$ are non-empty and disjoint.*

▶ **Theorem 7.** *If a (r, d)-hyperbolic graph G contains a $t \times t$ grid as a minor, then $t = O(\log(V(G))$.*

Proof. Without loss of generality we can assume $t = 2k + 1$. Such a $t \times t$ grid contains k cycles around the center of the grid. Let $v_c \in V$ be the vertex corresponding to this center. We have k cycles $C_1, \ldots, C_k$ in the graph V, each of which surrounds v_c and the preceding cycles. We use the following lemma:

▶ **Lemma 8.** *Every point in the drawing of cycle C_i is in distance $\Omega(i)$ from v. (The drawing is the polygon obtained as a union of the edges embedded in $\mathbb{H}^2$.)*

Therefore, $|C_i| \geq \mathrm{peri}(\Omega(i))/d$. Since peri grows exponentially, we get $k = O\log(|V|)$. ◀

Proof of Lemma 8. We will show that if every point in the drawing of C_i is in distance at least x from v, then every point in the drawing of C_{i+1} is in distance $x + c$ from v. By induction, this is enough to prove Lemma 8.

Figure 3 shows what happens for an Euclidean graph. (We define Euclidean graph just like hyperbolic graph except the differing underlying geometry.) In this picture, the drawing of C_i is a hexagon; cycle C_i has six vertices around v_c. For each of these seven vertices the exclusion zone of radius $r/2$ around it is shown (in green). It is clear from the picture (and the definition of Euclidean graph) that no point in the drawing of C_{i+1} may fall into the gray/green region. All points in distance $\leq r/2$ from C_i fall in the gray/green region, thus our claim is true for $c = h = r/2$.

In the hyperbolic plane the situation is slightly different. Because parallel lines work differently in the hyperbolic plane (they "diverge"), we have $c = h < r/2$, where h depends on r and d (we have $h = \Theta(re^{-d})$ for large values of d). Still, our claim holds with $c > 0$. ◀

▶ **Corollary 9.** *Given a planar graph $G = (V, E)$, it is possible to find a tree decomposition of width $O(\log|V|)$ in polynomial time.*

Proof. From Theorem 7 choose $t = O(\log|V|)$ such that G does not contain a $k \times k$ grid. From Theorem 4, it is possible to find a tree decomposition of width $5t - 6 = O(\log|V|)$. ◀

▶ **Corollary 10.** *HECSP (and thus HLCSP) can be solved in polynomial time.*

Proof. From Corollary 9 we can find a tree decomposition (V_T, E_T, B) of width $w = O(\log|V|)$. Then we use Dynamic Programming over (V_T, E_T). For every $b \in V_T$, we compute $s(w)$, the set of all possible colorings $c : X_b \in K$ such that there exists a coloring $c : X_b^+$ which extends c and which satisfies all the constraints on edges in X_b^+. This can be computed straightforwardly in every case from Lemma 6. The whole algorithm works in $O(|V| \cdot |K|^w)$, which is polynomial in $|V|$. ◀
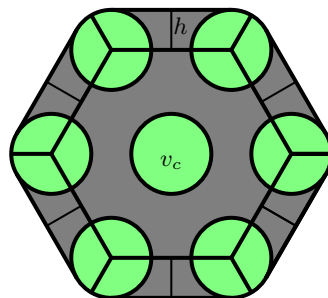
Figure 3 Cycles around v_c.

(a) two yellow.   **(b)** one group.   **(c)** two groups.   **(d)** four groups.
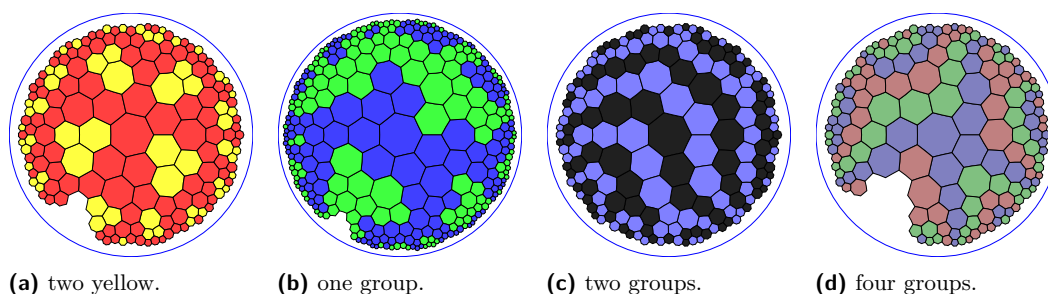
■ **Figure 4** Random colorings satisfying various constraints. Since the degree of our polynomial is quite high, these pictures took about a minute to make. Some cells have been removed from the full disk to reduce the treewidth.

6 Conclusion

We have shown that Minesweeper on hyperbolic tessellations can be solved in polynomial time. Our method is general: it works for any (r, d)-hyperbolic graph, and for any problem based on satisfying local constraints. Other than solving puzzles, this may be applied to procedural content generation. For example, the Wave Function Collapse (WFC) algorithm [4, 5] is used in procedurally generated games to procedurally generate maps satisfying local constraints (e.g., a mountain should not appear close to ocean, or roads should not branch nor end abruptly). In general finding out whether such constraints can be satisfied is NP-complete (although this does not happen in typical PCG applications). Our algorithm can be adjusted to count the number of satisfying colorings, and to produce one of them, randomly chosen (Figure 4).

Our results do not generalize to the three-dimensional hyperbolic space $\mathbb{H}^3$. This is because the Euclidean plane can be isometrically embedded in the three-dimensional hyperbolic space $\mathbb{H}^3$ as a horosphere. Minesweeper played on a tiling which tessellates such a horosphere into squares is NP-complete.

HyperRogue also lets the player play Minesweeper in bounded hyperbolic manifolds, such as the Klein quartic or other Hurwitz manifolds. Hurwitz manifolds are quotient spaces of $\mathbb{H}^2$ which can be tiled with regular heptagons with angles 120° (see Figure 1b), and that are maximally symmetric, i.e., there exists an isometry of the Hurwitz manifold M into itself which map any heptagon with any orientation into any heptagon with any orientation. Tessellations of quotient space are no longer planar, thus Theorem 4 nor our proof of Theorem 7 is no longer valid in quotient spaces of $\mathbb{H}^2$. Therefore, the complexity of Minesweeper on such manifolds does not follow from our results. There are less symmetric hyperbolic manifolds into which the Euclidean square grid, or even higher-dimensional Euclidean grids, can be embedded [13], and thus Minesweeper on them is NP-complete; however, highly symmetric manifolds have a more restricted structure.

───── **References** ─────

1 Sergio Bermudo, José M. Rodríguez, José M. Sigarreta, and Jean-Marie Vilaire. Gromov hyperbolic graphs. *Discrete Mathematics*, 313(15):1575–1585, 2013. `doi:10.1016/j.disc.2013.04.009`.

2 James W. Cannon, William J. Floyd, Richard Kenyon, Walter, and R. Parry. Hyperbolic geometry. In *In Flavors of geometry*, pages 59–115. University Press, 1997. Available online at `http://www.msri.org/communications/books/Book31/files/cannon.pdf`.

3 Alexander Grigoriev. Tree-width and large grid minors in planar graphs. *Discrete Mathematics & Theoretical Computer Science*, 13:13–20, January 2011.

4 Maxim Gumin. WaveFunctionCollapse, 2017. URL: `https://github.com/mxgmn/WaveFunctionCollapse`.

5 Isaac Karth and Adam M. Smith. WaveFunctionCollapse is Constraint Solving in the Wild. In *Proceedings of the 12th International Conference on the Foundations of Digital Games*, FDG '17, New York, NY, USA, 2017. Association for Computing Machinery. `doi:10.1145/3102071.3110566`.

6 Richard Kaye. Minesweeper is np-complete. *The Mathematical Intelligencer*, 22(2):9–15, March 2000. `doi:10.1007/BF03025367`.

7 Eryk Kopczyński, Dorota Celińska, and Marek Čtrnáct. Hyperrogue: Playing with hyperbolic geometry. In David Swart and Carlo H. Séquin and Kristóf Fenyvesi, editor, *Proceedings of Bridges 2017: Mathematics, Art, Music, Architecture, Education, Culture*, pages 9–16, Phoenix, Arizona, 2017. Tessellations Publishing. Available online at `http://archive.bridgesmathart.org/2017/bridges2017-9.pdf`.

8 John Lamping, Ramana Rao, and Peter Pirolli. A focus+context technique based on hyperbolic geometry for visualizing large hierarchies. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '95, pages 401–408, New York, NY, USA, 1995. ACM Press/Addison-Wesley Publishing Co. `doi:10.1145/223904.223956`.

9 Maurice Margenstern. Pentagrid and heptagrid: the fibonacci technique and group theory. *Journal of Automata, Languages and Combinatorics*, 19(1-4):201–212, 2014. `doi:10.25596/jalc-2014-201`.

10 Tamara Munzner. Exploring large graphs in 3d hyperbolic space. *IEEE Computer Graphics and Applications*, 18(4):18–23, 1998. `doi:10.1109/38.689657`.

11 Fragkiskos Papadopoulos, Maksim Kitsak, M. Angeles Serrano, Marian Boguñá, and Dmitri Krioukov. Popularity versus Similarity in Growing Networks. *Nature*, 489:537–540, September 2012.

12 N. Robertson, P. Seymour, and R. Thomas. Quickly excluding a planar graph. *Journal of Combinatorial Theory, Series B*, 62(2):323–348, 1994. `doi:10.1006/jctb.1994.1073`.

13 Zeno Rogue. HyperRogue: Experiments with Geometry, 2020. URL: `http://www.roguetemple.com/z/hyper/geoms.php`.

14 Sci-Tech Binary Ltd. Co. Warped mines, 2019. URL: `https://apps.apple.com/br/app/hypersweeper/id1450066199`.