ASP-Completeness of Hamiltonicity in Grid Graphs, with Applications to Loop Puzzles

MIT Hardness Group[1]
CSAIL, Massachusetts Institute of Technology, Cambridge, MA, USA

Josh Brunner
CSAIL, Massachusetts Institute of Technology, Cambridge, MA, USA

Lily Chung
CSAIL, Massachusetts Institute of Technology, Cambridge, MA, USA

Erik D. Demaine
CSAIL, Massachusetts Institute of Technology, Cambridge, MA, USA

Della Hendrickson
CSAIL, Massachusetts Institute of Technology, Cambridge, MA, USA

Andy Tockman
CSAIL, Massachusetts Institute of Technology, Cambridge, MA, USA

Abstract

We prove that Hamiltonicity in maximum-degree-3 grid graphs (directed or undirected) is ASP-complete, i.e., it has a parsimonious reduction from every NP search problem (including a polynomial-time bijection between solutions). As a consequence, given k Hamiltonian cycles, it is NP-complete to find another; and counting Hamiltonian cycles is #P-complete. If we require the grid graph's vertices to form a full $m \times n$ rectangle, then we show that Hamiltonicity remains ASP-complete if the edges are directed or if we allow removing some edges (whereas including all undirected edges is known to be easy). These results enable us to develop a stronger "T-metacell" framework for proving ASP-completeness of rectangular puzzles, which requires building just a single gadget representing a degree-3 grid-graph vertex. We apply this general theory to prove ASP-completeness of 37 pencil-and-paper puzzles where the goal is to draw a loop subject to given constraints: Slalom, Onsen-meguri, Mejilink, Detour, Tapa-Like Loop, Kouchoku, Icelom; Masyu, Yajilin, Nagareru, Castle Wall, Moon or Sun, Country Road, Geradeweg, Maxi Loop, Mid-loop, Balance Loop, Simple Loop, Haisu, Reflect Link, Linesweeper; Vertex/Touch Slitherlink, Dotchi-Loop, Ovotovata, Building Walk, Rail Pool, Disorderly Loop, Ant Mill, Koburin, Mukkonn Enn, Rassi Silai, (Crossing) Ichimaga, Tapa, Canal View, and Aqre. The last 13 of these puzzles were not even known to be NP-hard. Along the way, we prove ASP-completeness of some simple forms of Tree-Residue Vertex-Breaking (TRVB), including planar multigraphs with degree-6 breakable vertices, or with degree-4 breakable and degree-1 unbreakable vertices.

2012 ACM Subject Classification Theory of computation → Problems, reductions and completeness

Keywords and phrases pencil-and-paper puzzles, computational complexity, parsimony

Digital Object Identifier 10.4230/LIPIcs.FUN.2024.23

Related Version *Full Version*: `https://arxiv.org/abs/2405.08377`

Acknowledgements This paper was initiated during open problem solving in the MIT class on Algorithmic Lower Bounds: Fun with Hardness Proofs (6.5440) taught by Erik Demaine in Fall 2023. We thank the other participants of that class for helpful discussions and providing an inspiring atmosphere. Some figures drawn using SVG Tiler [`https://github.com/edemaine/svgtiler`].

[1] Artificial first author to highlight that the other authors (in alphabetical order) worked as an equal group. Please include all authors (including this one) in your bibliography, and refer to the authors as "MIT Hardness Group" (without "et al.").

© MIT Hardness Group, Josh Brunner, Lily Chung, Erik D. Demaine, Della Hendrickson, and Andy Tockman;
licensed under Creative Commons License CC-BY 4.0

12th International Conference on Fun with Algorithms (FUN 2024).
Editors: Andrei Z. Broder and Tami Tamir; Article No. 23; pp. 23:1–23:20
LIPICS Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Hamiltonicity is one of the core NP-complete problems, used as the basis for countless NP-hardness reductions. It accounts for two of Karp's 21 NP-complete problems [22]: directed and undirected Hamiltonian cycle. It has been shown to remain NP-complete for many restricted graph classes: undirected maximum-degree-3 graphs [15], undirected bipartite graphs [24], undirected 3-connected 3-regular bipartite graphs [2], undirected 2-connected 3-regular bipartite planar graphs [2], undirected 3-connected 3-regular planar graphs of minimum face degree 5 [16], directed planar graphs with indegree and outdegree at most 2 and total degree at most 3 [29], and so on.

One of the most useful special cases of Hamiltonicity is (square) ***grid graphs***: graphs whose vertices are a subset of the 2D integer lattice, with an edge connecting two vertices exactly when they have distance 1. Itai, Papadimitriou, and Szwarcfiter [19] proved that Hamiltonicity is NP-complete in grid graphs. Papadimitriou and Vazirani [28] improved this result by proving Hamiltonicity NP-complete in grid graphs of maximum degree 3. Together, these results strengthen most of the special graph classes mentioned above (as grid graphs are necessarily planar and bipartite), with a stronger geometric guarantee. Other papers extend these results to other 2D grids [6, 10, 17]. Hamiltonicity in grid graphs is the foundation for NP-hardness proofs of countless games and puzzles, from video games [13, 9, 1] to pencil-and-paper puzzles [36, 3], as well as practical problems such as lawn mowing and milling [5, 4].

But what about ***parsimonious*** reductions that preserve the number of solutions? A particularly strong form of this notion is ASP-completeness: an NP search problem P is ***ASP-complete*** [37] if there is a polynomial-time reduction from every NP search problem Q to P along with a polynomial-time bijection converting every solution of P to a unique solution of Q and vice versa. If P is ASP-complete, then the decision version of P is NP-complete, counting solutions to P is #P-complete, and the ***k-ASP*** P problem – given an instance of P and k solutions, find another solution – is NP-complete for any $k \geq 0$ [37].

Only a few versions of Hamiltonicity are known to be ASP-complete, or weaker, #P-complete. Liśkiewicz, Ogihara, and Toda [25] proved #P-completeness of Hamiltonicity in undirected 3-regular planar graphs (based on [16]). Seta [30] proved ASP-completeness of Hamiltonicity in undirected maximum-degree-3 planar graphs (based on [29]). Bosboom et al. [8] proved ASP-completeness of Hamiltonicity in *directed* 3-regular (indegree 2 and outdegree 1 or vice versa) planar graphs (based on [29]). But what about grid graphs?

1.1 Our Results

In this paper, we prove that Hamiltonicity in maximum-degree-3 grid graphs is ASP-complete. Thus this popular problem can serve as a foundation for ASP-completeness proofs as well. The same result holds for Hamiltonicity in *directed* maximum-degree-3 grid graphs, where each edge has a specified direction. As mentioned above, grid graphs are bipartite and planar, so these results roughly strengthen the ASP-completeness results mentioned above, except that we can guarantee "maximum-degree-3" but not "3-regular". (No grid graphs are 3-regular; consider the top-left corner. Furthermore, undirected 3-regular graphs have an even number of Hamiltonian cycles by Smith's Theorem [34], so we cannot hope for ASP-completeness in this case: the 1-ASP decision problem is trivial, while the 1-ASP construction problem is in PPA [27].)

The basis for this result is another form of Hamiltonicity called ***Tree-Residue Vertex-Breaking (TRVB)*** [11], previously used to analyze Hamiltonicity in grid graphs [10]. In TRVB, we are given a graph where some vertices are breakable, and the goal is to ***break***

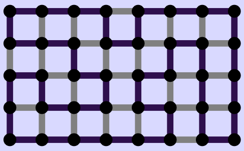
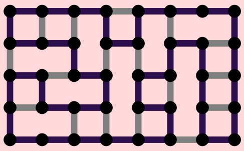
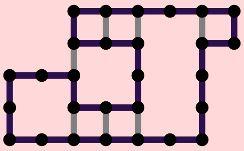
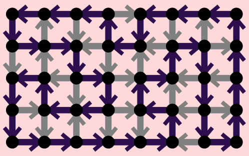
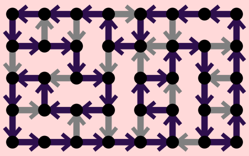
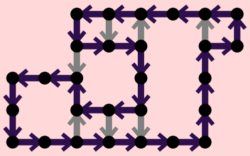
	Rectangular	Max-degree-3 spanning subgraph of rectangular	Max-degree-3
Undirected	P [19] 	ASP-complete [§4.2] 	ASP-complete [§4.3] 
Directed	ASP-complete [§4.1] 	ASP-complete [§4.2] 	ASP-complete [§4.3] 

Table 1 Complexity of Hamiltonicity in various types of grid graphs. Each cell shows an example of a Hamiltonian graph of the specified type, with a darkened Hamiltonian cycle. The first and third column concern true grid graphs, where there is an edge between each pair of vertices at distance 1. In the first and second columns, the vertices form exactly an $m \times n$ rectangle, whereas the third column allows an induced subgraph of a rectangular grid graph. The middle column concerns graphs constructed from a rectangular grid graph by removing some edges (but no vertices) so that each vertex has degree at most 3. The second and third columns have maximum degree 3.

a subset of the breakable vertices – replacing each broken degree-k vertex with k degree-1 vertices – to make the graph into a tree. This problem has a known characterization of what degrees of breakable or unbreakable vertices make the problem polynomial vs. NP-complete [11]. We prove that several forms of TRVB are in fact ASP-complete, including planar multigraphs with degree-6 breakable vertices, and planar multigraphs with degree-4 breakable and degree-1 unbreakable vertices.

We also study even more geometric forms of grid-graph Hamiltonicity. Suppose instead of allowing an arbitrary set of vertices on the square grid, we require the vertex set to be an entire $m \times n$ rectangle of integer points. Such graphs are known as ***rectangular grid graphs*** [19]. In this case, undirected Hamiltonicity is known to be easy [19]. But we show that *directed* Hamiltonicity in rectangular grid graphs is ASP-complete. Alternatively, if the graph is undirected but we allow removing some edges (but not vertices) from the rectangular grid – a spanning subgraph of a rectangular grid graph – then Hamiltonicity is also ASP-complete. Table 1 summarizes these results.

Rectangular grid graphs are useful because many (if not most) pencil-and-paper puzzles take place on a full rectangular grid. In particular, the ***T-metacell framework*** of Tang [32] shows how NP-hardness for a pencil-and-paper puzzle often follows from building a single gadget, essentially representing a degree-3 vertex that must be visited at least once. In Section 5, we extend this framework to prove ASP-completeness as well. We also extend the framework to allow for T-metacells where some exits are directed (usable in only one direction) and up to one exit is forced (must be used). In some cases, we need to build more than one T-metacell to handle different orientations of directions and/or forced edges.

Finally, in Section 6, we apply this framework to prove ASP-completeness of 37 pencil-and-paper puzzles, listed in Table 2. Five of these results use the same reduction from [32], while the remainder involve creating new T-metacell gadget(s). For thirteen of the analyzed puzzles, even our NP-hardness result is new.

Table 2 Our results on pencil-and-paper puzzles. All ASP-completeness results are new; some are via an existing reduction [32] and some are via a new reduction; and some puzzles were not even known to be NP-hard. (Puzzles known to be NP-hard have corresponding citations).

Games	#	New ASP-Hardness	New Reduction	New NP-Hardness
Slalom/Suraromu [21, 32], Onsen-meguri [32], Mejilink [32], Detour [31, 32], Tapa-Like Loop [32], Kouchoku [32], Icelom [32]	7	yes	no	no
Masyu [14, 32], Yajilin [18, 32], Nagareru [20, 32], Castle Wall [32], Moon or Sun [20, 32], Country Road [18, 32], Geradeweg [32], Maxi Loop [32], Mid-loop [32], Balance Loop [32], Simple Loop [19, 32], Haisu [31, 32], Reflect Link [32], Linesweeper [26]	14	yes	yes	no
Vertex/Touch Slitherlink, Dotchi-Loop, Ovotovata, Building Walk, Rail Pool, Disorderly Loop, Ant Mill, Koburin, Mukkonn Enn, Rassi Silai, (Crossing) Ichimaga, Tapa, Canal View, Aqre	16	yes	yes	yes

2 Connections Between Problems

We collect together some useful equivalences between problems on plane graphs, which are variously present in the literature [12, 11].

▶ **Definition 1** ([11]). *The **Tree-Residue Vertex-Breaking** (TRVB) problem takes place on an undirected multigraph with vertices marked as either "breakable" or "unbreakable". The goal is to* break *a subset S of the breakable vertices to leave a tree – to break a vertex of degree d, replace it with d new leaves attached to its incident edges. In other words, the graph obtained from G by subdividing every edge and deleting the vertices in S must be a tree.*

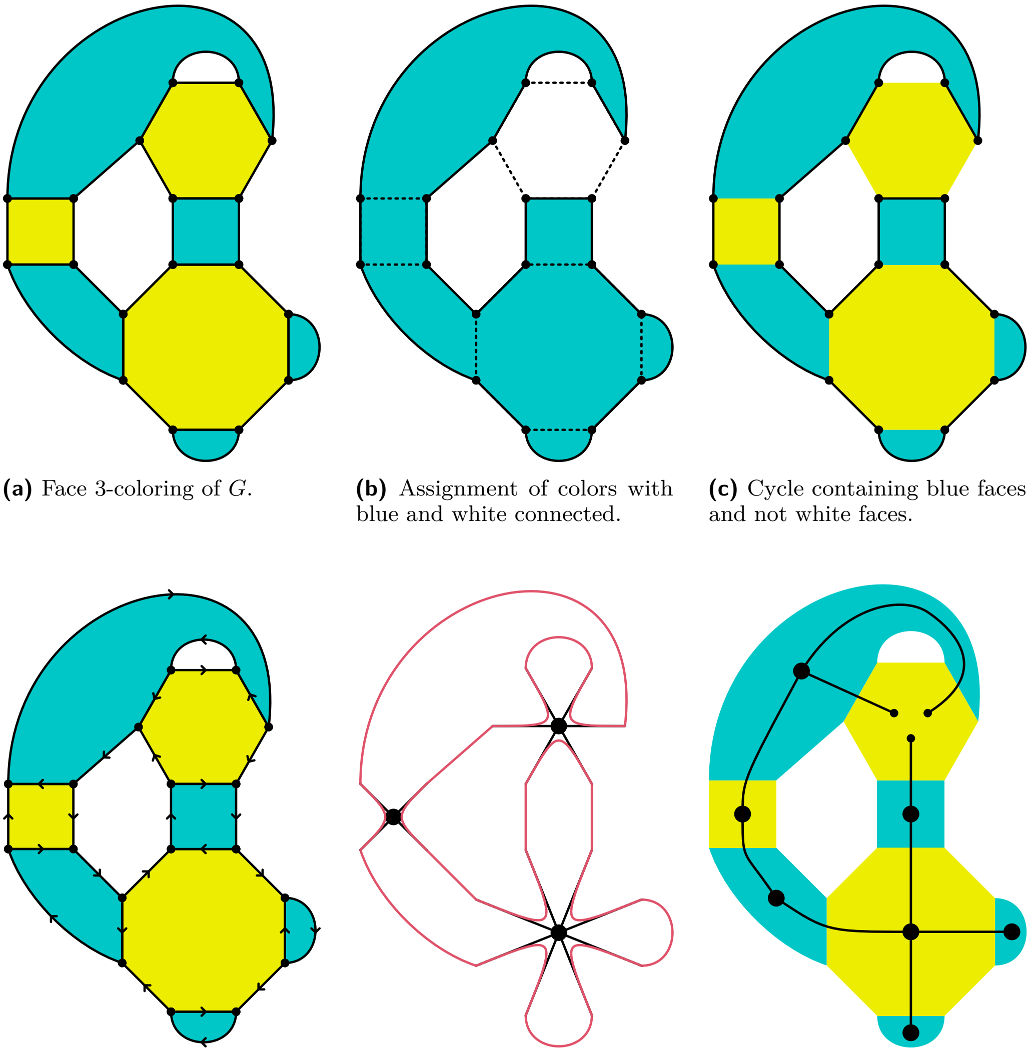
▶ **Definition 2** ([7, 12]). *Given a plane multigraph, a **kiki Euler tour** is a cycle which traverses every edge exactly once, such that any time the cycle enters a vertex via an edge e, it leaves by an edge adjacent to e in the cyclic order.*[2]

The following is a well-known result with a long history; see [33].

▶ **Theorem 3.** *Every Eulerian plane graph where every face is a triangle, except possibly the exterior face (a "near-triangulation"), has a proper vertex 3-coloring.*

Let G be a connected 3-regular bipartite plane multigraph, and let $\widetilde{G}$ be its plane dual. By Theorem 3, $\widetilde{G}$ is 3-colorable; equivalently it is possible to 3-color the faces of G so that adjacent faces have different colors, where faces are regarded as adjacent if they share an edge. Note that in such a 3-coloring, the three faces around a single vertex contain each color exactly once.

[2] This notion is one of two definitions of "nonintersecting" or "noncrossing Euler tour". We avoid this term to avoid confusion with the other definition, where an Euler tour is has a ***crossing*** if there are four edges e, e', f, f' adjacent to a single vertex so that e' follows e and f' follows f in the tour, and $\{e, e'\}$ alternates with $\{f, f'\}$ in the cyclic order [33]. Noncrossing Euler tours in this sense always exist, whereas kiki is a stricter condition.



(a) Face 3-coloring of G.

(b) Assignment of colors with blue and white connected.

(c) Cycle containing blue faces and not white faces.

(d) Directed graph G_1.

(e) Kiki Euler tour of G_2.

(f) Tree-Residue Vertex-Breaking of G_3.

Figure 1 Illustration of Lemma 4.

Let us fix such a coloring using the colors {white, blue, yellow} such that the exterior face is colored white. Define the following graphs:

- G_1 is the directed plane multigraph obtained from G by orienting every blue face clockwise and every white face counterclockwise. This fully determines the orientation.
- G_2 is the plane multigraph obtained from G by contracting every yellow face to a single vertex.
- G_3 is the subgraph of $\widetilde{G}$ induced by the non-white vertices.

▶ **Lemma 4.** *There are bijections between the following sets:*

(i) *Assignments of colors {white, blue} to each yellow vertex of $\widetilde{G}$ such that the white induced subgraph is connected and the blue induced subgraph is also connected.*

(ii) *Hamiltonian cycles of G which contain all blue faces and no white faces.*

(iii) *Hamiltonian cycles of G which use every edge separating white faces from blue faces.*

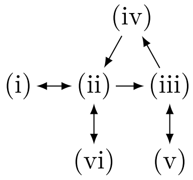


Figure 2 The bijections we define for Lemma 4.

(iv) *Directed Hamiltonian cycles of G_1.*

(v) *Kiki Euler tours of G_2.*

(vi) *Tree-Residue Vertex-Breakings of G_3, where yellow vertices are breakable and blue vertices are unbreakable.*

Proof. Refer to Figure 1. We give explicit transformations between the sets; it can be checked that these transformations invert each other as needed. Figure 2 summarizes the transformations we describe, which form a strongly connected graph.

(i) → (ii): Consider an assignment of colors to faces of G. For each vertex, two of the faces around it are one color and the third is the other color, so exactly two edges incident to it separate blue from white. The set of all edges separating blue from white thus forms a collection of cycles visiting each vertex once.
We claim that this is actually a single cycle. If it were multiple cycles, they would divide the plane into more than two regions. Two of those regions must be the same color (blue or white), violating the assumption that each color is connected.
So we have a Hamiltonian cycle separating blue from white, and since the exterior face is white, it contains all blue faces and no white faces of G.

(ii) → (i): Given a cycle, assign blue to exactly the faces it contains. Since the cycle is Hamiltonian, it does not intersect itself, so the blue faces are connected and the white faces are connected.

(ii) → (iii): If C contains all blue faces and no white faces, then it must use every edge separating white from blue.

(iii) → (iv): If C is a cycle on G_1 which uses every edge separating white from blue, then at each individual vertex it is impossible for C to reverse directions; thus it is always consistent with the orientations, so it is a directed Hamiltonian cycle.

(iv) → (ii): Suppose C is a directed Hamiltonian cycle of G_1. Since C visits every vertex, it contains at least one edge of every face. Because C contains an edge of the exterior face its orientation must be consistently clockwise. Therefore C it encounters every blue face on its right side and every white face on the left, meaning it contains every blue face and does not contain any white faces.

(iii) → (v): The edges separating white and blue faces are exactly the edges of G_3 remaining after contracting the yellow faces. Let C be a Hamiltonian cycle of G containing every white-blue edge, and let C' be the Euler tour of G_3 obtained from C by the contraction. It must be the case that C contains exactly half of the edges incident to each yellow face, each of which connects two adjacent white-blue edges; so C' is kiki.

(v) → (iii): Suppose C' is a kiki Euler tour of G_3. Let C be the set of edges of G consisting of all white-blue edges, together with those that connect consecutive edges in C'; then C is a Hamiltonian cycle of G containing every white-blue edge.

(ii) → (vi): Note that G_3 does not have any edges between two breakable vertices, so breaking a vertex is equivalent to removing it and all incident edges. Thus TRVB becomes "find an induced subgraph of G_3 containing all unbreakable vertices which is a tree".

Given a cycle C, break all yellow vertices which are outside C, or equivalently take the induced subgraph on vertices inside C. This subgraph is clearly connected. If it has a cycle, there is a face of $\widetilde{G}$ inside that cycle, which corresponds to a vertex v of G. Then v is strictly inside C. But v must touch a white face, contradicting the fact that all white faces are outside C. Hence the induced subgraph on vertices inside C is a tree.

(vi) → (ii): Take C to be the boundary of the tree containing blue faces and nonbroken yellow faces. Then C is a cycle because it bounds a tree, its interior contains all blue faces (which cannot be broken) and no white faces (which are not present in G_3. Finally, C is Hamiltonian because every vertex is incident to an edge separating blue from white, which must be in C. ◀

Furthermore, given any of the graphs G_i, equivalents to the others can be obtained by analogous transformations. So these various problems can be regarded as equivalent.

An important special case of TRVB is when every breakable vertex has degree at most 3. For planar graphs this condition is equivalent to requiring that every yellow face of the graph G in the preceding discussion is a digon or triangle; it is also equivalent to kiki Euler tour with vertices of degree at most 6. In this case, the problem can be solved in polynomial time by reducing it to a matroid parity problem.[12][11] In the next section we will discuss breakable vertices with higher degrees, with which the problem turns out to be ASP-complete.

3 ASP-Completeness of Tree-Residue Vertex-Breaking

Demaine and Rudoy [11] prove several NP-hardness results for TRVB using reductions from finding Hamiltonian cycles on a max-degree-3 planar directed graph. At the time, this Hamiltonian cycle problem was not known ASP-complete, so they did not consider ASP-completeness.

More recently, Bosboom et al. [8] showed that finding Hamiltonian cycles on a directed max-degree-3 planar graph is ASP-complete, using a reduction from positive 1-in-3SAT.

Several of the reductions used by Demaine and Rudoy [11] are easily verified to be parsimonious, proving ASP-completeness. We are specifically interested in the results of Section 4, on planar $(\{k\},\{4\})$-TRVB.

They first reduce finding Hamiltonian cycles on a max-degree-3 planar directed graph to finding Hamiltonian cycles on a planar graph where all vertices have indegree and outdegree 2 and vertices have their two in-edges and their two out-edges adjacent in the planar embedding. This last condition is called ***non-alternating***, because vertices are not allowed to alternate in-edges and out-edges. The reduction is by contracting forced edges, and is straightforwardly parsimonious.

▶ **Theorem 5.** *Finding Hamiltonian cycles on non-alternating indegree-2 outdegree-2 planar graphs is ASP-complete.*

Next, Demaine and Rudoy reduce this problem to a version of Tree-Residue Vertex-Breaking. Specifically, Demaine and Rudoy [11] prove NP-hardness of TRVB on a planar graph where each unbreakable vertex has degree 4 and each breakable vertex has degree k, for any constant $k \geq 4$. This is ***planar $(\{k\},\{4\})$-Tree-Residue Vertex-Breaking***. This reduction is a bit more complicated (see Section 4.2 and in particular Figures 11 through 13

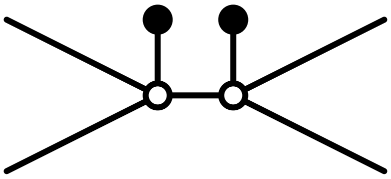


Figure 3 Simulating a degree-4 unbreakable vertex using degree-4 breakable vertices (white) and degree-1 unbreakable vertices (black).

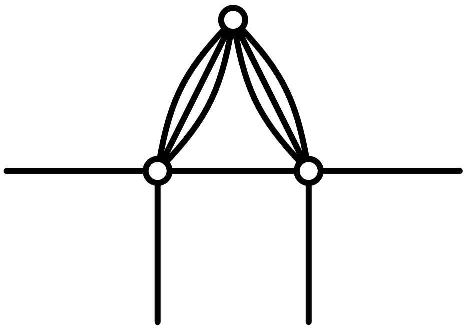


Figure 4 Simulating a degree-4 unbreakable vertex using degree-6 breakable vertices.

of [11]) but it is again parsimonious; indeed, [11, Lemmas 4.14 and 4.15] show that there is a bijection between Hamiltonian cycles in the input problem and solutions to the TRVB instance.

▶ **Theorem 6.** *Planar $(\{k\}, \{4\})$-TRVB is ASP-complete, for each $k \geq 4$.*

To further simplify our reductions, we will use a slightly simpler version of TRVB: degree-4 breakable vertices and degree-1 unbreakable vertices.

▶ **Theorem 7.** *Planar $(\{4\}, \{1\})$-TRVB is ASP-complete.*

Proof. It suffices to parsimoniously simulate a degree-4 unbreakable vertex. Such a simulation is shown in Figure 3. No vertex in the simulation can be broken in a solution to TRVB. ◀

▶ **Theorem 8.** *Planar $(\{6\}, \emptyset)$-TRVB is ASP-complete.*

Proof. It again suffices to simulate a degree-4 breakable vertex. Such a simulation is shown in Figure 4. If the top vertex is not broken, both others must be broken, disconnecting the middle edge. So the top vertex must be broken, and then the other two vertices must not be. ◀

4 Hamiltonian Cycles in Grid Graphs

In this section, we prove ASP-completeness of finding Hamiltonian cycles in several natural classes of grid graphs. We begin by defining the types of graph that appear in our results.

▶ **Definition 9.** *A **grid graph** is an induced subgraph of the square lattice. That is, its vertices are a subset of $\mathbb{Z}^2$, and it has an edge between each pair of vertices at distance 1. In a **directed grid graph**, each edge has a direction, so there is exactly one edge between each pair of vertices at distance 1.*

▶ **Definition 10.** *A **rectangular** grid graph is one whose vertex set consists of all lattice points within a rectangle.*

▶ **Definition 11.** *A graph is **max-degree-3** if each of its vertices have degree at most 3.*

▶ **Definition 12.** *A **spanning subgraph** of G is a subgraph of G which contains all of the vertices (and some subset of the edges) of G.*

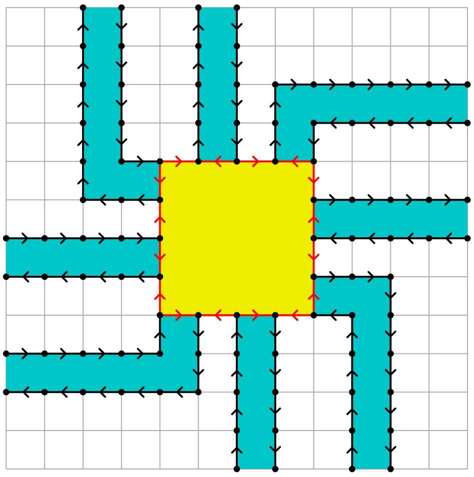


Figure 5 An example showing how reductions from TRVB to Hamiltonian cycle work.

Note that grid graphs contain all possible edges: graphs that contain only some of the edges are (spanning) subgraphs of grid graphs.

We consider three types of graph for each of undirected and undirected. Our results are summarized in Table 1.

Most of our ASP-completeness results are by reductions from planar $(\{4\}, \{1\})$-TRVB, and use the same core idea illustrated in Figure 5. This is a breakable degree-8 vertex, with the yellow square in the middle representing the vertex itself and the blue tentacles representing edges. We replace every vertex in the TRVB instance with a vertex like the one shown, and connect the tentacles of adjacent vertices. By Lemma 4, Hamiltonian cycles of the resulting graph correspond to solutions of the original TRVB instance.

This idea works equally well for directed and undirected graphs. To apply this idea to each of the five types of graph we prove ASP-completeness for, we need to show how to draw gadgets for degree-4 breakable and degree-1 unbreakable vertices in that type of graph, while ensuring that the tentacles representing edges do not interfere with each other.

4.1 Rectangular Grid Graphs

▶ **Theorem 13** ([19]). *Finding Hamiltonian cycles on an undirected rectangular grid graph is in P.*

▶ **Theorem 14.** *Finding Hamiltonian cycles on a directed rectangular grid graph is ASP-complete.*

Proof. We first consider directed grid graphs, and later fill in holes to make them rectangular. Everything we need for this is shown in Figure 6. The yellow rectangles are degree-4 breakable vertices with exactly two local solutions, and the dead end in the bottom left is a degree-1 unbreakable vertex. As before, blue is inside the loop and yellow might be inside the loop depending on the choice made for a vertex gadget. If we ignore the gray edges, this is essentially the same as Figure 5.

We just need to ensure that gray edges cannot be used, which we can do by orienting them carefully. Ignoring the H-shaped construction in the center for the moment, each black edge is either the only edge pointing towards or the only edge pointing away from some vertex (depending on which side of the tentacle it's on), and thus must be used in a Hamiltonian

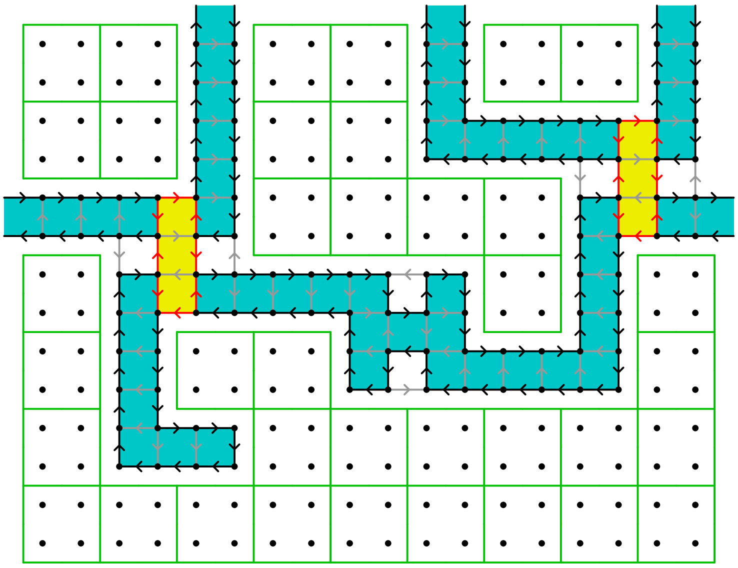


Figure 6 TRVB gadgets for directed grid graphs, showing two breakable degree-4 vertices connected by an edge and an unbreakable degree-1 vertex.

cycle. We call such an edge ***forced***. Each gray edge (still ignoring the H) shares either its source or its target with a black edge, and thus cannot be used. We call such an edge ***unusable***.

This requires the orientation of the gray edges relative to a tentacle to be different on the two ends of the tentacle, which is what the H achieves: one can verify by repeatedly finding forced edges and deleting unusuable edges that any Hamiltonian cycle must use all black edges and no gray edges in the H. Each tentacle representing an edge between two degree-4 breakable vertices will have such an H.

This reduction proves a weaker version of the theorem: Finding Hamiltonian cycles on a directed grid graph is ASP-complete. It remains to fill all of the unused space to make a rectangular grid graph.

If we place each vertex gadget, H, and turn on the same parity, the construction lies neatly on a 2×2 grid, and in particular the holes are made of 2×2 squares. Figure 6 indicates these squares in green. In addition, in each hole at least one of these squares is adjacent to a forced edge: all black edges except a few in each H are forced,[3] and each hole is adjacent to a non-H section of tentacle provided we do not use any extremely short tentacles.

Pick one such 2×2 square, and add four new vertices to fill it. Assume that the adjacent forced edge is the only outgoing edge from its source; the case where it is the only edge pointing towards its target is similar but with directions reversed. This situation is illustrated in Figure 7 (left), with the forced edge in blue. Now reverse the forced edge, and add new edges as shown on the right of Figure 7 (omitting any edges between a vertex in the square and a vertex outside it which doesn't yet exist). It is straightforward to check that all gray edges are unusable, so any Hamiltonian cycle must follow the blue path, which is equivalent to the original forced edge but consumes the added vertices.

[3] They all become forced after deleting some unusable edges, but it's simpler to argue that hole filling works with directly forced edges.

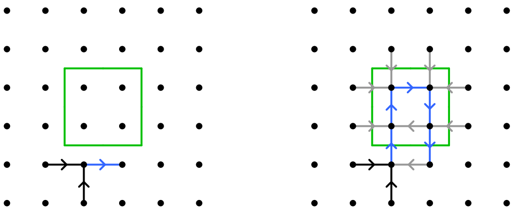


Figure 7 Filling holes in a directed rectangular grid graph.

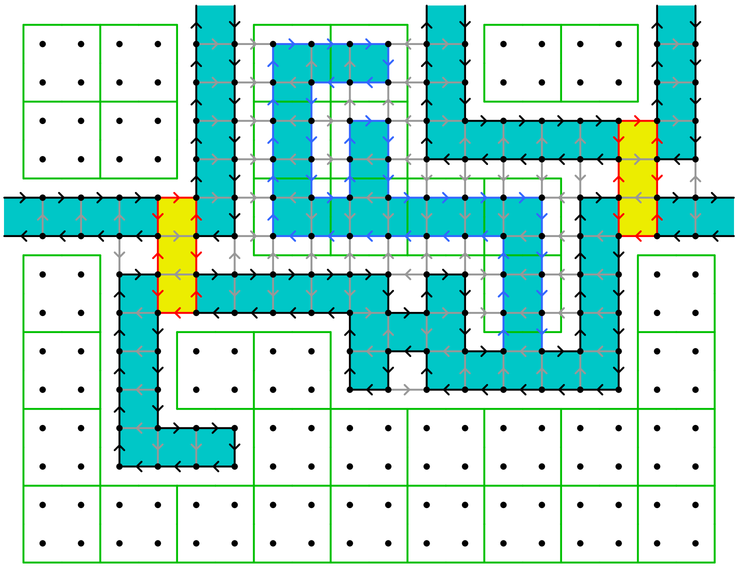


Figure 8 Figure 6 after some hole filling.

Filling this small portion of hole preserves the fact that every hole has a 2 × 2 square adjacent to a forced edge, since the three relevant blue edges are forced. Thus we can repeat this process until all holes are filled, ultimately filling each hole with paths that outline a spanning forest of the 2 × 2 squares. Figure 8 shows what this looks like after filling (the visible portion of) the top middle hole in Figure 6.

The result is a directed rectangular grid graph which is equivalent to the original directed grid graph for the purposes of Hamiltonian cycles. Hence Hamiltonian cycles in the final graph correspond to solutions to the instance of TRVB. ◀

4.2 Max-Degree-3 Spanning Subgraphs of Rectangular Grid Graphs

▶ **Theorem 15.** *Let G be a directed max-degree-3 spanning subgraph of a rectangular grid graph. Consider the promise problem of finding an* undirected *Hamiltonian cycle on G, subject to the promise that all such cycles respect the given edge directions; that is, they would also be valid directed Hamiltonian cycles of G. This promise problem is ASP-complete.*

Proof. We modify the construction from Theorem 14 by simply removing all of the gray edges. Inspection of Figure 8 reveals that every vertex is incident to at most three non-gray edges: vertices along tentacles have two forced edges, and vertices in degree-4 vertex gadgets have one forced edge and two optional red edges. Filling holes preserves the non-gray degree of existing vertices and adds vertices with two non-gray edges.

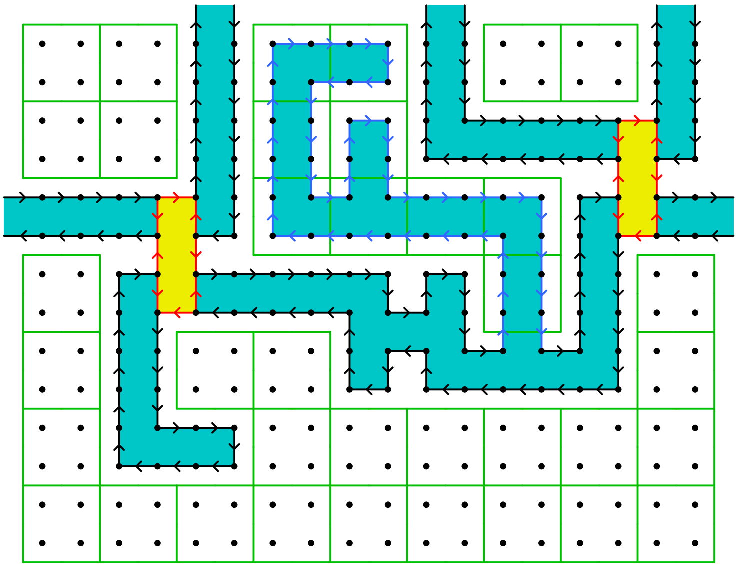


Figure 9 Figure 8 after removing gray edges.

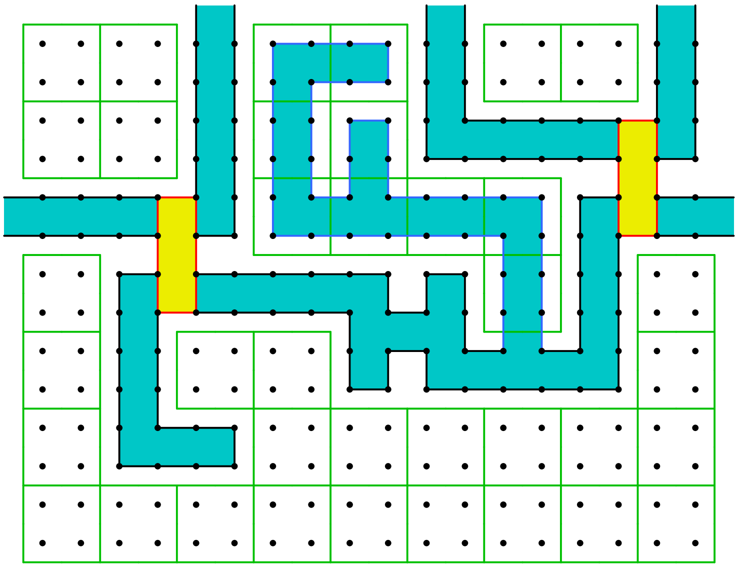


Figure 10 Figure 9 after forgetting directions of edges.

In the previous proofs, all of the possible solutions only used non-gray edges. Thus, we can adapt the previous reduction by simply deleting all gray edges, obtaining a directed max-degree-3 spanning subgraph of a rectangular grid graph. For instance, doing this to Figure 8 yields Figure 9, which also has the advantage of being easier to read.

By the proof of Lemma 4, directed Hamiltonian cycles on G are the same as undirected Hamiltonian cycles on G, and the set of such cycles is in bijection with solutions of the original TRVB instance. ◀

▶ **Corollary 16.** *Finding Hamiltonian cycles on a directed max-degree-3 spanning subgraph of a rectangular grid graph is ASP-complete.*

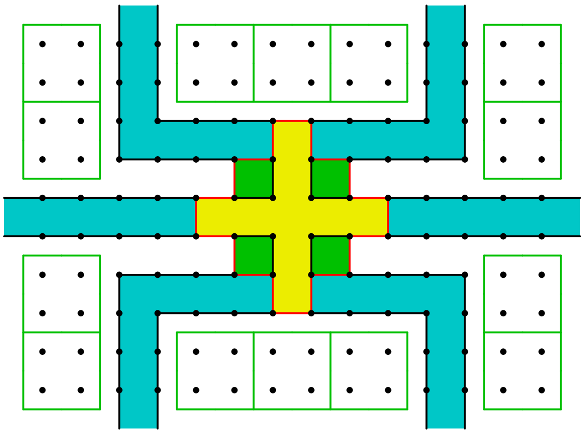


Figure 11 A breakable degree-6 TRVB vertex gadget for undirected max-degree-3 spanning subgraphs of rectangular grid graphs.

Proof. This is a special case of Theorem 15. ◀

In the undirected case, we can strengthen the assumption about forced edges. For undirected graphs, an edge is ***forced*** if it is incident to a degree-2 vertex, since both edges incident to such a vertex must be used in any Hamiltonian cycle. A degree-3 vertex in a subgraph of a grid graph has two edges in opposite directions, which we call ***side*** edges, and a third edge between them, which we call the ***center*** edge. In this case, we can assume not only that each degree-3 vertex has a forced edge, but that this forced edge is a side edge, further reducing the number of distinct vertices we need to simulate for an application.

▶ **Theorem 17.** *Finding Hamiltonian cycles on an undirected max-degree-3 spanning subgraph of a rectangular grid graph is ASP-complete, even when every degree-3 vertex has a forced side edge.*

Proof. We are not able to directly build breakable degree-4 TRVB vertices under these constraints. However, we are able to build a breakable degree-6 vertex, so we reduce from planar $(\{6\}, \emptyset)$-TRVB, which was shown ASP-complete in Theorem 8.

Our breakable degree-6 vertex gadget is shown in Figure 11. Black edges are forced, and red edges are optional. Note that vertices in tentacles all have degree 2, and each degree-3 vertex inside the vertex gadget has a forced side edge. This is equivalent to the cycle of red edges turning at every vertex. The vertex gadget has exactly two local solutions, which each use alternating red edges.

As before, blue tentacles are inside the cycle, and the yellow region is inside the cycle in one of the local solutions, corresponding to not breaking the TRVB vertex. We have new color as well: the green squares are inside the cycle in the other solution, when the TRVB vertex is broken. It is clear by inspection that the yellow local solution connects all six tentacles, and the green local solution disconnects them all.

Finally, we connect vertex gadgets along tentacles and fill holes in exactly the same way as before. Filling holes uses only degree-2 vertices, so it does not introduce degree-3 vertices without forced side edges. ◀

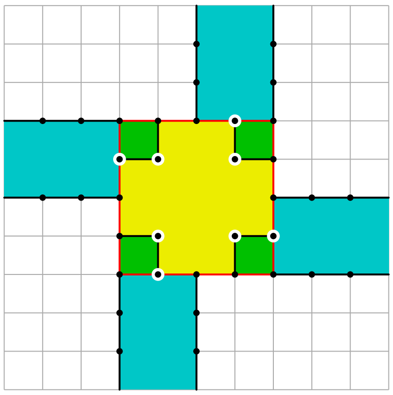


Figure 12 A breakable degree-4 TRVB vertex gadget for undirected max-degree-3 grid graphs. Removing the vertices highlighted in white gives an unbreakable degree-4 vertex gadget.

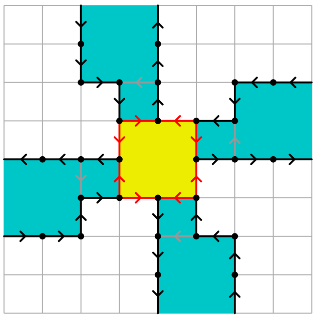


Figure 13 A breakable degree-4 TRVB vertex gadget for directed max-degree-3 grid graphs.

4.3 Max-Degree-3 Grid Graphs

▶ **Theorem 18.** *Finding Hamiltonian cycles on an undirected max-degree-3 grid graph is ASP-complete, even when every vertex has a forced edge.*

Proof. This proof is sketched, and its key gadget is shown, by Demaine and Rudoy [11], but at the time TRVB was not known to be ASP-complete, so it was purely a simpler proof of NP-hardness used to motivate the usefulness of TRVB.

Like most of our other proofs, we reduce from planar $(\{4\},\{1\})$-TRVB. Our breakable degree-4 vertex gadget is shown in Figure 12. The main difficulty in this case is that we need the paths on each side of a tentacle to be separated by distance at least 2, so that the cycle cannot cross between the two sides (and all tentacle edges are forced). As usual, black edges are forced, and there are exactly two solutions which each use alternating red edges. One solution puts the green region inside the cycle, and one puts the yellow region inside the cycle, corresponding to breaking and not breaking the vertex, respectively.

A degree-1 unbreakable vertex can be made by simply "capping off" a tentacle. Alternatively, we could reduce from $(\{4\},\{4\})$-TRVB, and construct a degree-4 unbreakable vertex gadget by removing the vertices highlighted in white from Figure 12. ◀

▶ **Theorem 19.** *Finding Hamiltonian cycles on a directed max-degree-3 grid graph is ASP-complete, even when every vertex has a forced edge.*

Proof. The proof is extremely similar to the previous proof. We again reduce from $(\{4\},\{1\})$-TRVB. Our degree-4 breakable vertex gadget is shown in Figure 13, and a degree-1 unbreakable vertex can again be made by capping off a tentacle. Black edges are forced and gray edges are unusable. We again keep the sides of a tentacle apart from each other (away from vertex gadgets) so that a cycle cannot leak between them.

As before, there are exactly two solutions to the vertex gadget, one of which put the yellow square inside the cycle corresponding to leaving the TRVB vertex unbroken. ◀

5 T-Metacells

Many puzzle genres which involve drawing a single loop are proven hard using reductions from various forms of grid graph Hamiltonicity. Tang [32] described a simple "T-metacell" framework for proving NP-hardness of these puzzles using grid graph Hamiltonicity. A

T-metacell is a gadget which represents a single degree-3 vertex in a grid graph. Each T-metacell is a (usually square) tile with 3 exits (on 3 of the 4 sides) such that the loop may traverse the gadget between any pair of exits. The gadget should be reflectable and rotatable, and the loop may travel between adjacent T-metacells only when both have exits along their shared border. Finally, the loop must be required to visit every T-metacell.

It's straightforward to see how T-metacells can simulate degree-3 vertices in a Hamiltonicity reduction; Tang showed that they can also simulate degree-2 vertices. Let G be a subgraph of a grid graph in which every vertex has degree 2 or 3. Degree-3 vertices of G can be replaced directly with T-metacells. To handle degree-2 vertices, consider the graph H on the same vertex set as G which has an edge between two lattice-adjacent vertices precisely when G is missing that edge. Then H consists of degree-1 and degree-2 vertices. Orient the edges of H into directed paths and cycles such that each vertex has a maximum indegree and outdegree of 1. Each degree-2 vertex of G can now be replaced by a T-metacell with its extra edge facing in the direction of the outward-pointing edge from that vertex in H. This ensures that this extra exit will always be facing a non-exit in the adjacent cell, so only the intended edges of G may be used by the loop.

We apply our results from Section 4 to show that solving T-metacell problems is ASP-complete, instead of just NP-hard. We extend the framework to allow for some exits of a T-metacell to be ***directed***, meaning that the loop must have a consistent orientation which agree with the directions of the exits it uses. We also allow for T-metacells to have one ***forced exit*** through which the loop must pass. Note that when all three exits are directed, these necessarily create a forced exit: there must be either a lone exit directed inwards or a lone exit directed outwards, which in either case must be chosen. T-metacells with forced edges can be classified into two categories: symmetric and asymmetric. A symmetric T-metacell has its two unforced edges directly opposite each other, while an asymmetric T-metacell has its two unforced edges adjacent. We use this classification to reduce the number of distinct gadgets which need to be constructed to apply the framework.

▶ **Corollary 20.** *Finding Hamiltonian cycles on a rectangular grid of undirected T-metacells is ASP-complete.*

Proof. We reduce from finding Hamiltonian cycles on max-degree-3 spanning subgraphs of rectangular grid graphs (Theorem 17). Replace each vertex with a undirected T-metacell, handling degree-2 vertices as described above. ◀

▶ **Corollary 21.** *Finding Hamiltonian cycles on a rectangular grid of required-edge directed T-metacells is ASP-complete.*

Proof. We reduce from finding Hamiltonian cycles on directed max-degree-3 spanning subgraphs of rectangular grid graphs (Corollary 16). Place a T-metacell for each degree-3 vertex, and handle degree-2 vertices in the same way as above. The direction of the unusable edge on a T-metacell at a degree-2 vertex can be arbitrary. ◀

▶ **Corollary 22.** *Finding Hamiltonian cycles on a rectangular grid of asymmetric required-edge undirected T-metacells is ASP-complete.*

Proof. In the proof of Theorem 17, every degree-3 vertex conveniently has a forced side edge, which is equivalent to being a asymmetric undirected T-metacell. Degree-2 vertices require a bit more care, but are not an obstruction: after deciding how to orient T-metacells as described above, note that for each degree-2 vertex, at least one of its edges is a side edge of the T-metacell. So we can simply place a T-metacell with that side edge forced. ◀

▶ **Corollary 23.** *Finding Hamiltonian cycles on a rectangular grid of required-edge directed asymmetric T-metacells and required-edge undirected symmetric T-metacells is ASP-complete.*

Proof. We reduce from the promise problem of finding a Hamiltonian cycle of a directed max-degree-3 spanning subgraph of a rectangular grid graph, with the promise that every undirected Hamiltonian cycle is a valid directed Hamiltonian cycle (Theorem 15). We perform the same replacement of vertices with T-metacells as in Corollary 21, except that the symmetric T-metacells are undirected. We claim that Hamiltonian cycles of the original graph are in bijection with solutions to the T-metacell instance. A directed Hamiltonian cycle of the original graph clearly solves the T-metacell instance, since it correctly passes through the directions on the directed T-metacells. On the other hand, a solution to the T-metacell instance is necessarily an undirected Hamiltonian cycle of the original graph; by the promise, directed Hamiltonian cycles and undirected Hamiltonian cycles are the same. ◀

6 Applications

We apply our improved T-metacell framework to a variety of pencil-and-paper logic puzzles implemented by the online puzzle-solving interface "puzz.link" [23]. This web resource implements more than 240 different logic puzzles. It includes most genres published by the Japanese publisher Nikoli, whose puzzles have a long history of analysis from a computational complexity perspective [30] [37] [3] [35] [26] [32], as well as many others in a similar style.

We improve existing NP-hardness results for pencil-and-paper logic puzzles to ASP-completeness, and give new ASP-completeness results. Many of the ASP-completeness proofs consist of just a single T-metacell, demonstrating the ease of applying the framework for proving ASP-completeness. The main additional requirement when designing a T-metacell gadget for ASP-completeness proofs is that it be "parsimonious": for each pair of exits, there must be a *unique* local solution where the loop passes through those exits.

Full explanations for each proof can be found in the full version of this paper; due to space constraints, we present an abridged gallery of reductions here.

Figure 14 shows the gadgets for improving prior NP-hardness results to ASP-completeness, most of which consist of minor adjustments to existing T-metacells in [32] to ensure parsimony. We also make similar improvements for Yajilin, Moon and Sun, and Simple Loop via direct reductions from Hamiltonicity.

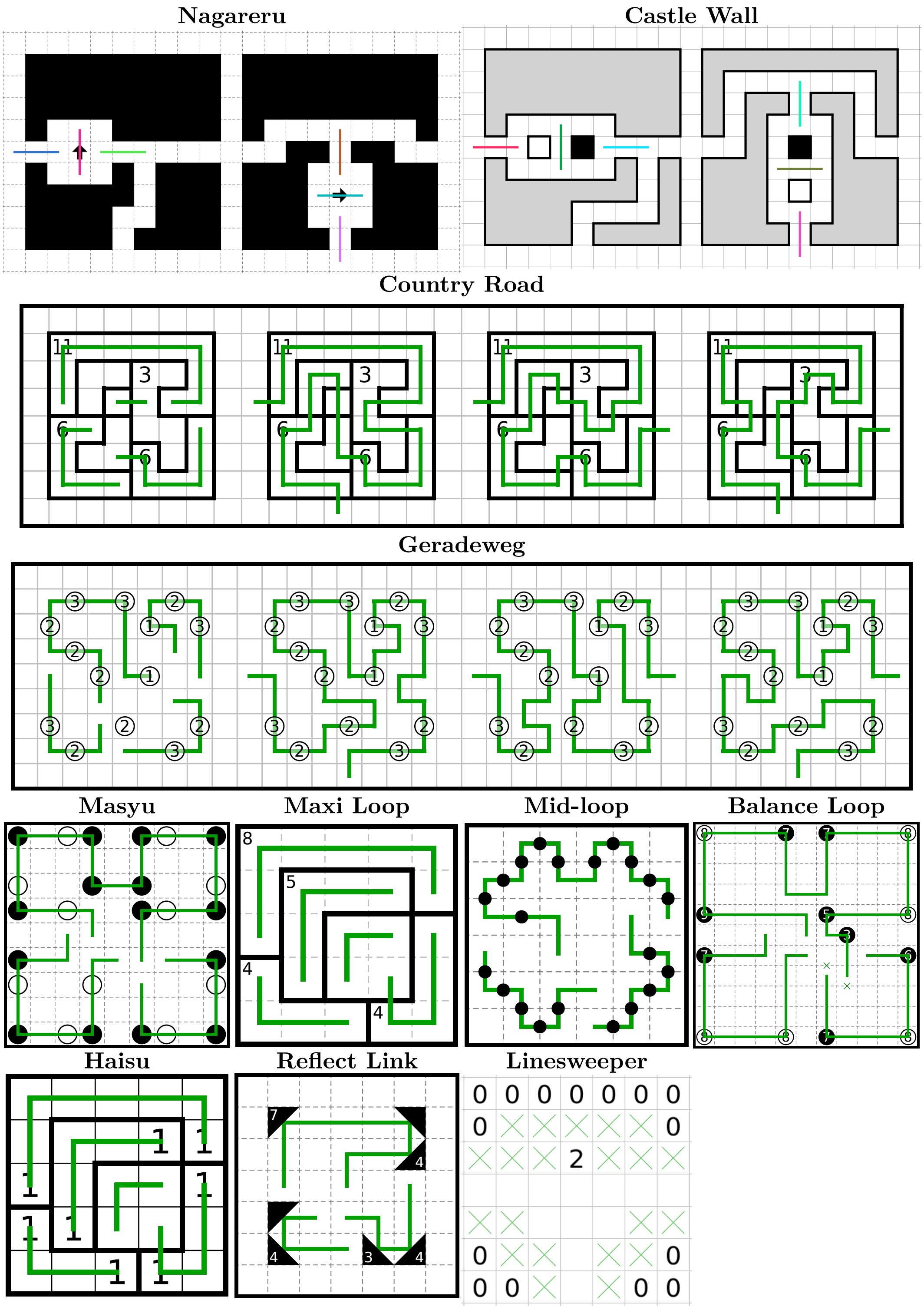
Figure 15 shows the gadgets for new NP- and ASP-completeness reuslts. We also give similar results for Dotchi Loop, Ovotovata, and Koburin via direct reductions from Hamiltonicity.

Finally, some puzzle genres were proved NP-complete by Tang, but we have not yet found parsimonious adaptations of the corresponding T-metacells. These genres are Angle Loop, Double Back, Scrin, Icebarn, and Icelom 2.

References

1 Zachary Abel, Jeffrey Bosboom, Michael Coulombe, Erik D. Demaine, Linus Hamilton, Adam Hesterberg, Justin Kopinsky, Jayson Lynch, Mikhail Rudoy, and Clemens Thielen. Who witnesses The Witness? Finding witnesses in The Witness is hard and sometimes impossible. *Theoretical Computer Science*, 839:41–102, November 2020.

2 Takanori Akiyama, Takao Nishizeki, and Nobuji Saito. NP-completeness of the Hamiltonian cycle problem for bipartite graphs. *Journal of Information Processing*, 3(2):73–76, 1980. URL: `https://ipsj.ixsq.nii.ac.jp/ej/?action=repository_action_common_download&item_id=59994&item_no=1&attribute_id=1&file_no=1`.



Nagareru

Castle Wall

Country Road

Geradeweg

Masyu

Maxi Loop

Mid-loop

Balance Loop

Haisu

Reflect Link

Linesweeper

Figure 14 T-metacells proving ASP-completeness of puzzles previously only known to be NP-complete.

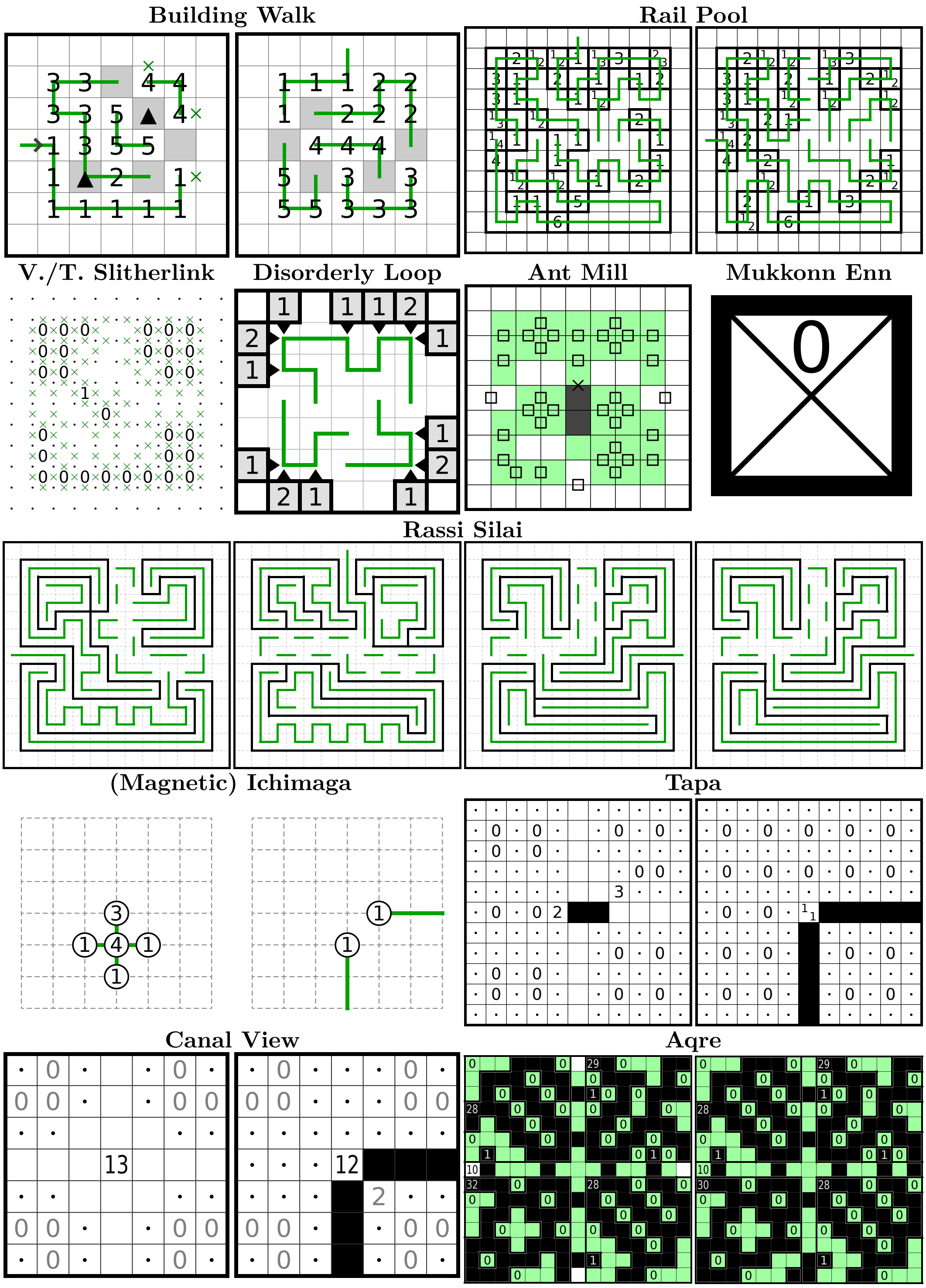


Figure 15 T-metacells proving ASP-completeness about puzzles whose complexity had not been studied.

3 Daniel Andersson. Hashiwokakero is NP-complete. *Information Processing Letters*, 109(19):1145–1146, 2009. doi:10.1016/j.ipl.2009.07.017.

4 Esther M. Arkin, Michael A. Bender, Erik D. Demaine, Sándor P. Fekete, Joseph S. B. Mitchell, and Saurabh Sethia. Optimal covering tours with turn costs. *SIAM Journal on Computing*, 35(3):531–566, 2005.

5 Esther M. Arkin, Sándor P. Fekete, and Joseph S.B. Mitc hell. Approximation algorithms for lawn mowing and milling. *Computational Geometry: Theory and Applications*, 17(1–2):25–50, 2000. doi:10.1016/S0925-7721(00)00015-8.

6 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. doi:10.1016/j.comgeo.2008.11.004.

7 Samuel W. Bent and Udi Manber. On non-intersecting Eulerian circuits. *Discrete Applied Mathematics*, 18(1):87–94, 1987. doi:10.1016/0166-218X(87)90045-X.

8 Jeffrey Bosboom, Charlotte Chen, Lily Chung, Spencer Compton, Michael Coulombe, Erik D. Demaine, Martin L. Demaine, Ivan Tadeu Ferreira Antunes Filho, Dylan Hendrickson, Adam Hesterberg, Calvin Hsu, William Hu, Oliver Korten, Zhezheng Luo, and Lillian Zhang. Edge matching with inequalities, triangles, unknown shape, and two players. *Journal of Information Processing*, 28:987–1007, 2020.

9 Erik D. Demaine, Joshua Lockhart, and Jayson Lynch. The computational complexity of Portal and other 3D video games. In *Proceedings of the 9th International Conference on Fun with Algorithms (FUN 2018)*, pages 19:1–19:22, La Maddalena, Italy, June 2018.

10 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. URL: https://arXiv.org/abs/1706.10046.

11 Erik D. Demaine and Mikhail Rudoy. Tree-Residue Vertex-Breaking: a new tool for proving hardness. In *Proceedings of the 20th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT 2018)*, pages 32:1–32:14, Malmö, Sweden, June 2018. Full paper at arXiv:1706.07900.

12 Tomás Feder and Carlos Subi. On Barnette's conjecture. *Electronic Colloquium on Computational Complexity (ECCC)*, January 2006.

13 Michal Forišek. Computational complexity of two-dimensional platform games. In *Proceedings of the 5th International Conference on Fun with Algorithms*, pages 214–227, Ischia, Italy, June 2010. doi:10.1007/978-3-642-13122-6_22.

14 Erich Friedman. Pearl puzzles are NP-complete. Manuscript, August 2002. URL: https://erich-friedman.github.io/papers/pearl.pdf.

15 M. R. Garey, D. S. Johnson, and L. Stockmeyer. Some simplified NP-complete problems. In *Proceedings of the 6th Annual ACM Symposium on Theory of Computing*, pages 47–63, Seattle, Washington, 1974. doi:10.1145/800119.803884.

16 M. R. Garey, D. S. Johnson, and R. Endre Tarjan. The planar Hamiltonian circuit problem is NP-complete. *SIAM Journal on Computing*, 5(4):704–714, 1976. doi:10.1137/0205049.

17 Kaiying Hou and Jayson Lynch. The computational complexity of finding Hamiltonian cycles in grid graphs of semiregular tessellations. In Stephane Durocher and Shahin Kamali, editors, *Proceedings of the 30th Canadian Conference on Computational Geometry (CCCG 2018)*, pages 114–128, Winnipeg, Canada, August 2018. URL: http://www.cs.umanitoba.ca/~cccg2018/papers/session3B-p1.pdf.

18 Ayaka Ishibashi, Yuichi Sato, and Shigeki Iwata. NP-completeness of two pencil puzzles: Yajilin and Country Road. *Utilitas Mathematica*, 88, June 2012. URL: https://utilitasmathematica.com/index.php/Index/article/view/863.

19 Alon Itai, Christos H. Papadimitriou, and Jayme Luiz Szwarcfiter. Hamilton paths in grid graphs. *SIAM Journal on Computing*, 11(4):676–686, 1982. doi:10.1137/0211056.

20 Chuzo Iwamoto and Tatsuya Ide. Moon-or-Sun, Nagareru, and Nurimeizu are NP-complete. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 105(9):1187–1194, 2022.

21 Shohei Kanehiro and Yasuhiko Takenaga. Satogaeri, Hebi, and Suraromu are NP-complete. In *Proceedings of the 3rd International Conference on Applied Computing and Information Technology/2nd International Conference on Computational Science and Intelligence*, pages 46–51, 2015.

22 Richard M. Karp. Reducibility among combinatorial problems. In *Proceedings of a Symposium on the Complexity of Computer Computations*, pages 85–103, Yorktown Heights, New York, March 1972. doi:10.1007/978-1-4684-2001-2_9.

23 Daisuke Kobayashi, Robert Vollmert, Lennard Sprong, et al. puzz.link. URL: https://puzz.link.

24 M. S. Krishnamoorthy. An NP-hard problem in bipartite graphs. *SIGACT News*, 7(1):26, January 1975. doi:10.1145/990518.990521.

25 Maciej Liśkiewicz, Mitsunori Ogihara, and Seinosuke Toda. The complexity of counting self-avoiding walks in subgraphs of two-dimensional grids and hypercubes. *Theoretical Computer Science*, 304(1):129–156, 2003. doi:10.1016/S0304-3975(03)00080-X.

26 Mieke Maarse. The NP-completeness of some lesser known logic puzzles. Bachelor's thesis, Utrecht University, June 2019. URL: https://studenttheses.uu.nl/handle/20.500.12932/33867.

27 Christos H. Papadimitriou. On the complexity of the parity argument and other inefficient proofs of existence. *Journal of Computer and System Sciences*, 48(3):498–532, 1994. doi:10.1016/S0022-0000(05)80063-7.

28 Christos H. Papadimitriou and Umesh V. Vazirani. On two geometric problems related to the travelling salesman problem. *Journal of Algorithms*, 5(2):231–246, 1984. doi:10.1016/0196-6774(84)90029-4.

29 Ján Plesník. The NP-completeness of the Hamiltonian cycle problem in planar diag raphs with degree bound two. *Information Processing Letters*, 8(4):199–201, April 1979. doi:10.1016/0020-0190(79)90023-1.

30 Takahiro Seta. The complexities of puzzles, Cross Sum, and their Another Solution Problems (ASP). Senior thesis, University of Tokyo, 2002. URL: https://web.archive.org/web/20221007013910/www-imai.is.s.u-tokyo.ac.jp/~seta/paper/senior_thesis/seniorthesis.pdf.

31 Hadyn Tang. On the NP-completeness of satisfying certain path and loop puzzles. arXiv:2004.12849, 2020. URL: https://arXiv.org/abs/2004.12849.

32 Hadyn Tang. A framework for loop and path puzzle satisfiability NP-hardness results. arXiv:2202.02046, 2022. URL: https://arXiv.org/abs/2202.02046.

33 Mu-Tsun Tsai and Douglas B. West. A new proof of 3-colorability of Eulerian triangulations. *Ars Mathematica Contempornanea*, 4(1):73–77, 2011. doi:10.26493/1855-3974.193.8E7.

34 W. T. Tutte. On Hamiltonian circuits. *Journal of the London Mathematical Society, Series 1*, 21(2):98–101, 1946. doi:10.1112/jlms/s1-21.2.98.

35 Akihiro Uejima, Hiroaki Suzuki, and Atsuki Okada. The complexity of generalized pipe link puzzles. *Journal of Information Processing*, 25:724–729, 2017.

36 Takayuki Yato. On the NP-completeness of the Slither Link puzzle. *IPSJ SiG Notes*, AL-74:25–32, 2000.

37 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. URL: http://ci.nii.ac.jp/naid/110003221178/en/.