Regular Paper

Selected Slither Link Variants are NP-complete

Jonas Kölker[1,a]

Abstract: In a Slither Link puzzle, the player must draw a cycle in a planar graph, such that the number of edges incident to a set of clue faces equals the set of given clue values. We show that for a number of commonly played graph classes, the Slither Link puzzle is **NP**-complete.

Keywords: puzzles, combinatorics, computational complexity, NP-completeness

1. Introduction and Definitions

Slither Link is a popular puzzle, published both in book form [4], as online static games [2] and as downloadable applications [6]. In its most general form, the rules can be formalized as follows.

Let $G = (V, E)$ be a plane graph [*1] and F_c a set of faces in G, the *clues*, and let $v: F_c \to \mathbb{N}$ be a function giving the *clue values*. The Slither Link decision problem is this: decide whether there is a set of edges $C \subseteq E$ such that C is a cycle and such that for all faces $F \in F_c$ the number of edges in C incident to F equals $v(F)$.

The original and most popular form of this puzzle is where V is a rectangular subset of $\mathbb{Z}^2$, e.g., $V = \{1, \dots, w\} \times \{1, \dots, h\}$, and E is the set of vertex pairs with Euclidean distance 1. This was shown to be **NP**-complete in Ref. [8]; a proof is readily available online in Ref. [7]. We show that for a small handful of other graph classes $\mathcal{G}$, the Slither Link decision problem restricted to $\mathcal{G}$ is also **NP**-complete.

For two of the graph classes, the triangular and hexagonal graphs, the **NP**-completeness is already known [5]. In fact, the authors show the problem to be ASP-complete on those graphs: given a solution, it is **NP**-hard to find another solution. These solution and hardness concepts are studied in Ref. [7]. One benefit of our reductions on those same graphs are their simplicity, which they achieve by relying on a result published after Ref. [5].

2. Membership of NP

First, we want to show that in the general form, i.e., for the class of all planar graphs (given as adjacency lists with clue faces and values given as edge lists and binary integers), the Slither Link decision problem is in **NP**. The argument is simple. Let C be given and consider the graph $G' = (V_C, C)$ where V_C is the set of endpoints of edges in C. It can easily be checked that each vertex v has degree 2 in G' and that G' is connected, and therefore a cycle. Checking that the number of edges incident to a face F of G equals $v(F)$ is likewise easy: compute a planar embedding in

linear time [1]; for each given clue, check that the given edge list actually is the boundary of a face, and that C overlaps that face F at exactly $v(F)$ edges.

For more specific Slither Link decision problems, i.e., for narrower graph classes $\mathcal{G}$, their **NP**-membership can be seen by the above proof and that testing membership of $\mathcal{G}$ is easy, if this is indeed the case. We will typically represent members of $\mathcal{G}$ not by an adjacency list but by parameters (similar to how w and h characterise rectangular grid graphs) from which an adjacency list can efficiently be computed. Then membership of $\mathcal{G}$ is given by construction and the above proof still applies.

3. The General Approach to Proving Hardness

It is known that the Hamiltonian Cycle problem is **NP**-complete on hexagon grids [3]: that is, given a set of points P on a hexagonal grid with a set of edges induced by unit Euclidean distances, the problem of determining whether a cycle exists that goes through every point exactly once is **NP**-complete. This is true even if the largest Euclidean distance between two points is required to be polynomial in the number of vertices in the graph; this is readily apparent from the proof in the citation. Thus, we can take the coordinates of hex grid points to be given in unary.

We will reduce this problem (Unary Hex Grid Hamiltonian Cycle) to each of our concrete Slither Link variants by showing how to construct vertex and non-vertex gadgets corresponding to points in P and not in P, respectively, and how edge gadgets are formed when vertex gadgets are combined. Lastly, we'll prove that this construction actually captures the behavior of vertices and edges in a hex grid Hamiltonian cycle problem, such that local solutions produce global solutions to Slither Link, which match Hamiltonian cycle solutions.

Let us be specific about hexagon grid graphs: let $x = (1, 0) \in \mathbb{R}^2$ and $w = (\cos 120°, \sin 120°) \in \mathbb{R}^2$, and let $T = \{ax + bw \mid a, b \in \mathbb{Z}\}$ be the *triangle grid* or lattice formed by integer combinations. The reader may recognize this as the Eisenstein integers. A *triangle grid graph* is a graph where V is a finite subset of T and E is those vertex pairs with Euclidean distance 1. The *hexagon grid* is the set $T_h = \{ax + bw \in T \mid a + b \not\equiv 0 \pmod 3\}$. Note that (x, w) is a basis for $\mathbb{R}^2$ so (a, b) is uniquely determined for every

---

[1] Department of Computer Science, Aarhus University, Denmark
[a] epona@cs.au.dk
[*1] A planar graph which not only can be but also *is* embedded in the plane.

point in T, and so any property of e.g., $a + b$ is a well-defined property of points in T. If one draws line segments between adjacent points in T_h, one will have drawn a hexagon tesselation of the plane. A *hexagon grid graph* is a graph induced by a finite set $V \subseteq T_h$ and unit distance edges.

Let $G_r = (T, E_1)$ be an infinite grid graph, e.g., T or T_h, and $G = (P, E)$ be a finite graph on this grid, i.e., $P \subseteq T$ and $E = (P \times P) \cap E_1$. Then we define $\text{Neigh}(P) := \{v \in T \mid \exists v' \in P : (v, v') \in E_1\}$ and $\text{Rim}(P) = \text{Neigh}(P) \setminus P$. Note that $P \cup \text{Rim}(P)$ has height and width only $O(1)$ larger than P, and can easily be computed.

The motivation for this definition is that if non-vertex gadgets are placed in a Slither Link graph, corresponding to points in $\text{Rim}(P)$, any candidate solution C which is partially contained in P will be unable to cross $\text{Rim}(P)$ and so will be completely confined to P: it can't be on the outside as long as vertex gadgets contain a clue face with a positive value, because then C wouldn't satisfy that clue.

4. The Dodecahedron Graph Class is Hard

The dodecahedron grid [*2] looks like this: draw the hex grid and replace each vertex with an equilateral triangle, turning each hexagon into a regular dodecahedron. Equivalently, draw a row of dodecahedrons next to one another with overlapping vertical edges; then, at each edge at five-o'-clock, draw a dodecahedron with the same edge in the eleven-o'-clock position. If the first row is w dodecahedrons wide and this row pair is repeated $\frac{h}{2}$ times vertically translated (plus a copy of the top row if h is odd), this is the *dodecahedron grid graph* [*3]. See also **Fig. 1**. They are represented by a pair (w, h) in unary, plus a clue/value list.

The Dodecahedronal Slither Link decision problem is the Slither Link decision problem, limited to this class of graphs. Now, let us consider a reduction from Unary Hex Grid Hamiltonian Cycle to Dodecahedronal Slither Link.

Let a Unary Hex Grid Hamiltonian Cycle problem $G = (P, E)$ be given. Note that we can biject the vertices in P onto triangular faces F_s on the dodecahedronal grid such that $(s, t) \in E$ if and only if F_s and F_t have an edge between them (i.e., between a corner in F_s and one in F_t). Let $v(F_s) = 1$ for $s \in P$ and $v(F_s) = 0$ for $s \in \text{Rim}(P)$, and undefined elsewhere (i.e., the set of clues is $P \cup \text{Rim}(P)$). With the hex grid coordinates in P given in unary, we can output w and h within polynomial time if we just choose them no larger than necessary to contain all the faces in this bi-
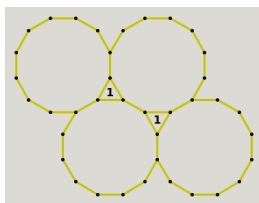

Fig. 1 2 by 2 dodecahedron grid, showing two vertex gadgets.

[*2] More commonly known as the truncated hexagonal tiling; we hope the reader will accept our idiosyncratic terminology.

[*3] We focus on tesselations of the plane by dodecahedrons and triangles in this particular grid structure, because this is the kind of graph produced by at least one popular Slither Link implementation.

jection, which is easy to do. If G has only a single vertex, we don't do the above. Instead, we output a particular yes-instance, the $w = h = 1$ puzzle with a single face, containing the clue 12.

The triangular faces with clue value 1 are the vertex gadgets; those with clue value 0 are the non-vertex gadgets, and their connecting edges are edge gadgets.

Theorem 1. *The Dodecahedronal Slither Link decision problem is NP-complete.*

Proof. The problem is clearly in **NP**—checking dodecahedron class membership is done by construction, so the previous proof applies. Thus, we only need to show that the reduction works.

If G has only a single vertex, it has a hamiltonian cycle and clearly the reduction outputs a yes-instance, so assume that G has multiple vertices.

The local solution to a vertex gadget is that C contains one edge of the triangular face and two edge gadgets adjacent to that face. Each face has three solutions, rotationally symmetric, corresponding to each choice of edge pairs.

If there is a hamiltonian cycle H in G, for every edge $e \in H$, include its corresponding edge gadget in C. This is consistent with exactly one local solution to every vertex gadget. Include the edges from the local solutions to vertex gadgets in C as well, and no more edges. Then C is a cycle because H is, and C satisfies all clue value constraints by construction.

On the other hand, any solution C will have to go through the gadget of every vertex in P (or it would violate a clue), and each vertex gadget will have two adjacent edge gadgets contained in C. This corresponds exactly to a hamiltonian cycle in G.

The result of the reduction is polynomial in size—the input has unary coordinates—and is easily (polynomial time) computable. □

5. The Triangular Graph Class is Hard

Recall the above definition of triangular grid graphs. We want to focus on a subset of these which has a very regular row/column structure. A row of length w is $2w$ triangles, alternating between pointing down and up, overlapping in the non-horizontal edges. In a graph with h rows, $h > 1$, each row is the horizontal mirror image of its predecessor. See **Fig. 2** for a $w = h = 6$ graph. We represent these graphs by unary encodings of w and h, as before, plus clue faces and values. We want to reduce Unary Hex Grid Hamiltonian Cycle to Triangular Slither Link.

Vertex gadgets look as in Fig. 2, or as in **Fig. 3** when partially solved—light grey edges are not a part of any solution, black edges are part of all solutions, and yellow edges are ambiguous.

Note that the displayed gadget corresponds to a hex grid vertex with edges going west, northeast and southeast. Mirror images of this gadget do the opposite. Non-vertex gadgets are as vertex gadgets except with all 1-valued clues replaced by 0-valued clues. Vertex gadgets are joined together as in **Fig. 5**. Note how the partial (and full) solutions are mirror images of one another.

To reduce a Unary Hex Grid Hamiltonian Cycle instance $G = (P, E)$ to Triangular Slither Link, put a vertex gadget in a triangle grid for each $v \in P$ and a non-vertex gadget for each $u \in \text{Rim}(P)$, joined as shown. Put these with as small coordinates as possible,
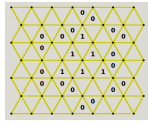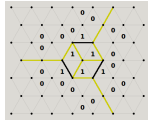
Fig. 2   Triangular vertex gadget.

Fig. 3   Partially solved vertex gadget.
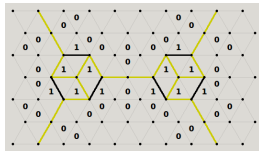

Fig. 4   Completely solved vertex gadget.
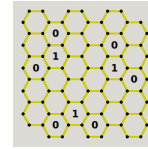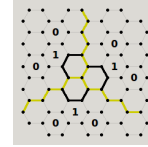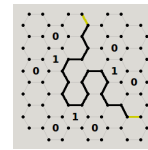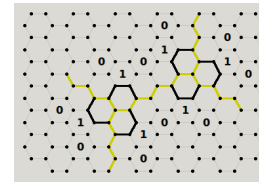

Fig. 5   Combined triangular vertex gadgets.

so as to minimise w and h.

Theorem 2. *The Triangular Slither Link decision problem is NP-complete.*

Proof.   We can establish **NP**-membership by the above observation: checking the graph in adjacency list form is easy, and it's easy to create that representation. Any pair of (w, h) is a member of our special class of triangle grid graphs.

Clearly the reduction is polynomial time: the gadgets are small and easily manipulable, and $w + h$ is small even in unary.

The local solution to a vertex gadget is shown in **Fig. 4**; this solution has two rotationally symmetric variants not shown, and each of the three corresponds to a choice of two selected edges incident to the gadget's corresponding vertex. There is also a fourth solution, a closed hexagonal loop, which corresponds to no edges being chosen. The local solution to a non-vertex gadget is one with no edges in the cycle.

If $|P| = 1$, then G has a trivial hamiltonian cycle, and the closed loop local solution is also a global solution.

If $|P| > 1$ then the fourth solution can't occur as part of any (global) solution: let C_4 be the edges in the hexagonal loop of a particular vertex v. Then any superset of C_4 can't be a cycle, and C_4 fails to satisfy the clues in the gadgets of $v' \in P$, $v' \neq v$.

Let a cycle H in G be given. Construct a solution C to the Triangular Slither Link instance by choosing local solutions to vertex gadgets corresponding to the choice of incident edges in H. This violates no clue, and the local solutions are consistent in their overlap (i.e., no edge is both a member and non-member of C) by construction. Globally, C is guaranteed to be a cycle (and not a multi-cycle cover) because H is a cycle.

Now, let a solution C to the Triangular Slither Link instance be given. Since the vertex gadgets only have local solutions corresponding to consistent choices of edges, and these edges globally

Fig. 6   Hexagonal vertex gadget.

Fig. 7   Partially solved vertex gadget.

Fig. 8   Completely solved vertex gadget.

Fig. 9   Combined hexagonal vertex gadgets.

form a cycle, we can produce a cycle H in G.   □

6.   The Hexagonal Graph Class is Hard

Recall the definition of the hexagonal grid. Again we look at a particular shape: we have w columns of h hexagons, each column's starting hexagon alternately to the southeast or northeast of its left hand neighbour. See **Fig. 7** for an example with $w = 6$ and $h = 5$. We want to show that the Hexagonal Slither Link problem is **NP**-complete.

The vertex gadgets look like in **Fig. 6** and its counterpart in **Fig. 9**. The non-vertex gadgets are the same, except with all clue values replaced by 0. The gadgets join as in Fig. 9. The reduction is as previous: put vertex gadgets at P-members and non-vertex gadgets at Rim(P), making w and h as small as possible. Note though, that we have to rotate our input graph 90°, essentially flipping the two axis, as our gadgets have edge gadgets going north/south rather than east/west.

Theorem 3. *The Hexagonal Slither Link decision problem is NP-complete.*

Proof.   Just like previously, the gadgets are small and easily manipulable, so the reduction is polynomial time.

The local solution is as in **Fig. 8**, with three rotations in total, allowing each choice of two out of three edges, and a fourth solution—a loop around a cluster of three hexagons—which is only valid for one-vertex graphs, in which case it's again a global solution. And again, the local non-vertex solution allows no edges. Each (global) solution is consistent with a hamiltonian cycle of the given Unary Hex Grid Hamiltonian Cycle problem,
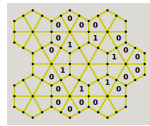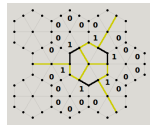
Fig. 10   Dual grid vertex gadget.

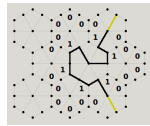**Fig. 11**   Partially solved vertex gadget.

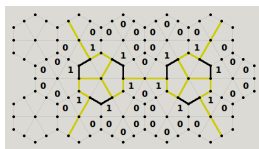**Fig. 12**   Completely solved vertex gadget.

Fig. 13   Combined dual vertex gadgets.

and corresponding to each hamiltonian cycle is a solution to the Slither Link problem.   □

7.   The Dual Graph Class is Hard

The triangular and hexagonal grids are each others' planar duals. The dual graph class [*4] arises from superimposing them on one another; that is, by adding vertices in the center of hexagons and connecting them to the centers of neighbouring hexagons, also adding a vertex when the connecting line intersects the edge between the connected hexagons. If we scale the unit down by 2, the hexagon centers have edges to all vertices in distance 2, while all other vertices have only unit distance edges. The particular structure we look at is a grid of h rows, each containing w hexagons—similar to the hexagonal grid, but rotated 90°.

The vertex gadget looks as in **Fig. 10** [*5]. If we make all possible local deductions, we arrive at **Fig. 11**. Non-vertex gadgets are vertex gadgets with all clues replaced by 0. Once again, put vertex gadgets at P and non-vertex gadgets at Rim(P), packed together to make w and h as small as possible; the combination looks as in **Fig. 13**.

Theorem 4.   *The Dual Slither Link decision problem is NP-complete.*

Proof.   Like above: the local solutions (**Fig. 12**) combine into a global solution, which matches hamiltonian cycles. Again, the single-vertex case has a special fourth local solution to vertex gadgets which is a hexagonal loop. This local solution is forbidden in global solutions to problems which the reduction yields on multi-vertex graphs.   □

[*4]   More commonly known as the deltoidal trihexagonal tiling; again we hope the reader will accept our idiosyncracy.

[*5]   The two leftmost empty hexagons are not part of the gadget proper, but an artifact of the software used to generate the illustration.

8.   Conclusion and discussion

We have shown how four Slither Link variants on particular graph classes are **NP**-complete. Note also that the reductions are all parsimonious—the number of solutions is preserved. This implies that the Slither Link variants are not only **#P**-complete if Unary Hex Grid Hamiltonian Cycle is, but also ASP-complete. To establish that Unary Hex Grid Hamiltonian Cycle is ASP-complete, one could try to find a chain of parsimonious reductions from one of the ASP-complete problems given in Ref. [7], for instance a SAT variety. The chain of known reductions showing the **NP**-completeness of Unary Hex Grid Hamiltonian Cycle might be parsimonious at all steps.

If one defines the Hamiltonian Cycle problem such that single-vertex graphs are no-instances, one can modify the presented reductions so they test for this and output a no-instance (for instance, give a face a positive clue value and all its neighbours the clue value 0).

Clearly Slither Link isn't difficult for all graph classes—it's rather easy to determine if G has a clue-satisfying cycle if G is a cycle. If one wants a deeper understanding of the hardness of Slither Link than we provide, one might try to find graph classes with non-obvious hardness characteristics.

References

[1]   Boyer, J.M. and Myrvold, W.J.: On the Cutting Edge: Simplified $O(n)$ Planarity by Edge Addition, *J. Graph Algorithms Appl.*, Vol.8, No.2, pp.241–273 (2004).

[2]   Bumgardner, J.: Free Slitherlink Puzzles from KrazyDad (online), available from ⟨http://www.krazydad.com/slitherlink/⟩ (accessed 2011-07-28).

[3]   Islam, K., Meijer, H., Rodríguez, Y.N., Rappaport, D. and Xiao, H.: Hamilton Circuits in Hexagonal Grid Graphs, *CCCG*, Bose, P. (Ed.), Carleton University, Ottawa, Canada, pp.85–88 (2007).

[4]   Nikoli: Rules of Slither Link, (online), available from ⟨http://www.nikoli.com/en/puzzles/slitherlink/rule.html⟩ (accessed 2011-07-28).

[5]   Nukui, K. and Uejima, A.: ASP-Completeness of the Slither Link Puzzle on Several Grids, *IPSJ SIG Notes*, Vol.2007, No.23, pp.129–136 (online), available from ⟨http://ci.nii.ac.jp/naid/110006249219/en/⟩ (accessed 2007-03-09).

[6]   Tatham, S.G.: Portable Puzzle Collection, (online), available from ⟨http://www.chiark.greenend.org.uk/˜sgtatham/puzzles/⟩ (accessed 2011-07-28).

[7]   Yato, T.: Complexity and completeness of finding another solution and its application to puzzles, Master's thesis, Graduate School of Science, the University of Tokyo (2003), available from ⟨http://www-imai.is.s.u-tokyo.ac.jp/˜yato/data2/MasterThesis.pdf⟩.

[8]   Yato, T. and Seta, T.: Complexity and Completeness of Finding Another Solution and Its Application to Puzzles, *IEICE Trans. Fundamentals*, Vol.E86-A, pp.1052–1060 (2003).

Jonas Kölker is a Ph.D. student at Aarhus University and a spare time free software developer.