Tree-Residue Vertex-Breaking: a new tool for proving hardness

Erik D. Demaine
MIT Computer Science and Artificial Intelligence Laboratory, 32 Vassar St., Cambridge, MA 02139, USA.
edemaine@mit.edu

Mikhail Rudoy
MIT Computer Science and Artificial Intelligence Laboratory, 32 Vassar St., Cambridge, MA 02139, USA. Now at Google Inc.
mrudoy@gmail.com
https://orcid.org/0000-0002-9210-1006

Abstract

In this paper, we introduce a new problem called Tree-Residue Vertex-Breaking (TRVB): given a multigraph G some of whose vertices are marked "breakable," is it possible to convert G into a tree via a sequence of "vertex-breaking" operations (replacing a degree-k breakable vertex by k degree-1 vertices, disconnecting the k incident edges)?

We characterize the computational complexity of TRVB with any combination of the following additional constraints: G must be planar, G must be a simple graph, the degree of every breakable vertex must belong to an allowed list B, and the degree of every unbreakable vertex must belong to an allowed list U. The two results which we expect to be most generally applicable are that (1) TRVB is polynomially solvable when breakable vertices are restricted to have degree at most 3; and (2) for any $k \geq 4$, TRVB is NP-complete when the given multigraph is restricted to be planar and to consist entirely of degree-k breakable vertices. To demonstrate the use of TRVB, we give a simple proof of the known result that Hamiltonicity in max-degree-3 square grid graphs is NP-hard.

We also demonstrate a connection between TRVB and the Hypergraph Spanning Tree problem. This connection allows us to show that the Hypergraph Spanning Tree problem in k-uniform 2-regular hypergraphs is NP-complete for any $k \geq 4$, even when the incidence graph of the hypergraph is planar.

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

Keywords and phrases NP-hardness, graphs, Hamiltonicity, hypergraph spanning tree

Digital Object Identifier 10.4230/LIPIcs.SWAT.2018.32

Related Version `https://arxiv.org/abs/1706.07900`

Acknowledgements We would like to thank Zachary Abel and Jayson Lynch for their helpful discussion about this research. We would also like to thank Yahya Badran for pointing out the connection between TRVB and the Hypergraph Spanning Tree problem.

1 Introduction

In this paper, we introduce the Tree-Residue Vertex-Breaking (TRVB) problem. Given a multigraph G some of whose vertices are marked "breakable," TRVB asks whether it is possible to convert G into a tree via a sequence of applications of the *vertex-breaking*

16th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT 2018).
Editor: David Eppstein; Article No. 32; pp. 32:1–32:14
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

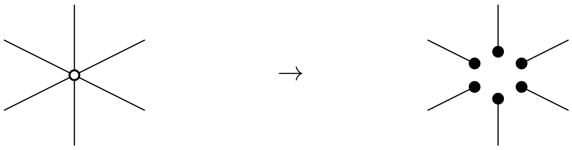

Figure 1 The operation of breaking a vertex. The vertex (left) is replaced by a set of degree-1 vertices with the same edges (right).

operation: replacing a degree-k breakable vertex with k degree-1 vertices, disconnecting the incident edges, as shown in Figure 1.

In this paper, we analyze the computational complexity of this problem as well as several variants (special cases) where G is restricted with any subset of the following additional constraints:

1. every breakable vertex of G must have degree from a list B of allowed degrees;
2. every unbreakable vertex of G must have degree from a list U of allowed degrees;
3. G is planar;
4. G is a simple graph (rather than a multigraph).

Modifying TRVB to include these constraints makes it easier to reduce from the TRVB problem to some other. For example, having a restricted list of possible breakable vertex degrees B allows a reduction to include gadgets only for simulating breakable vertices of those degrees, whereas without that constraint, the reduction would have to support simulation of breakable vertices of any degree.

We prove the following results (summarized in Table 1), which together fully classify the variants of TRVB into polynomial-time solvable and NP-complete problems:

1. Every TRVB variant whose breakable vertices are only allowed to have degrees of at most 3 is solvable in polynomial time.
2. Every planar simple graph TRVB variant whose breakable vertices are only allowed to have degrees of at least 6 and whose unbreakable vertices are only allowed to have degrees of at least 5 is solvable in polynomial time (and in fact the correct output is always "no").
3. In all other cases, the TRVB variant is NP-complete. In particular, the TRVB variant is NP-complete if the variant allows breakable vertices of some degree $k \geq 4$, and in the planar graph case, also allows either breakable vertices of some degree $b \leq 5$ or unbreakable vertices of some degree $u \leq 4$. For example, for any $k \geq 4$, TRVB is NP-complete in planar multigraphs whose vertices are all breakable and have degree k.

Among these results, we expect the most generally applicable to be the results that (1) TRVB is polynomially solvable when breakable vertices are restricted to have degree at most 3; and (2) for any $k \geq 4$, TRVB is NP-complete when the given multigraph is restricted to be planar and to consist entirely of degree-k breakable vertices.

Application to proving hardness

In general, the TRVB problem is useful when proving NP-hardness of what could be called *single-traversal problems*: problems in which some space (e.g., a configuration graph or a grid) must be traversed in a single path or cycle subject to local constraints. Hamiltonian Cycle and its variants fall under this category, but so do other problems. For example, a single traversal problem may allow the solution path/cycle to skip certain vertices entirely while mandating other local constraints. In other words, TRVB can be a useful alternative to Hamiltonian Cycle when proving NP-hardness of problems related to traversal.

Table 1 A summary of this paper's results (where B and U are the allowed breakable and unbreakable vertex degrees).

All breakable vertices have small degree ($B \subseteq \{1,2,3\}$)	**Graph restrictions**	**All vertices have large degree** ($B \cap \{1,2,3,4\} = \emptyset$ **and** $U \cap \{1,2,3,4,5\} = \emptyset$)	**TRVB variant complexity**	**Section**
Yes	*	*	Polynomial Time	Section 9
No	Planar or simple or unrestricted	*	NP-complete	Sections 4, 5, 6
No	Planar and simple	No	NP-complete	Section 7
No	Planar and simple	Yes	Polynomial Time (every instance is a "no" instance)	Section 8

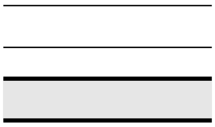

Figure 2 Abstraction of a possible edge gadget (top) and the local solution (bottom). The bold paths are (forced to be) part of the traversal while the "inside" of the gadget is shown in grey.

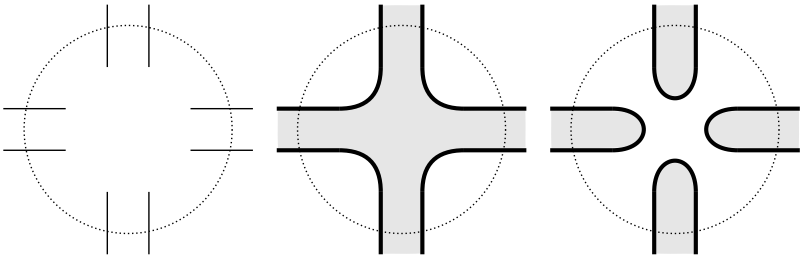

Figure 3 Abstraction of a possible breakable vertex gadget. The gadget should join some number of edge gadgets (in this case four) as shown on the left. The center and right figures show the two possible local solutions to the breakable vertex gadget. One solution connects the interiors of the incoming edge gadgets within the vertex gadget while the other disconnects them. In both figures, the bold paths are part of the traversal, while the "inside" of the gadget is shown in grey.

To prove a single-traversal problem hard by reducing from TRVB, it is sufficient to demonstrate two gadgets: an edge gadget and a breakable degree-k vertex gadget for some $k \geq 4$. This is because TRVB remains NP-hard even when the only vertices present are degree-k breakable vertices for some $k \geq 4$. Furthermore, since this version of TRVB remains NP-hard even for planar multigraphs, this approach can be used even when the single-traversal problem under consideration involves traversal of a planar space.

One possible approach for building the gadgets is as follows. The edge gadget should contain two parallel paths, both of which must be traversed because of the local constraints of the single-traversal problem (see Figure 2). The vertex gadget should have exactly two possible solutions satisfying the local constraints of the problem: one solution should disconnect the regions inside all the adjoining edge gadgets, while the other should connect these regions inside the vertex gadget (see Figure 3). We then simulate the multigraph from the input TRVB instance by placing these edge and vertex gadgets in the shape of the input multigraph as shown in Figure 4.

When trying to solve the resulting single-traversal instance, the only option (while satisfying local constraints) is to choose one of the two possible local solutions at each vertex gadget, corresponding to the choice of whether to break the vertex. The candidate solution

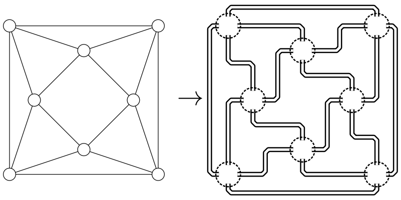

Figure 4 The input multigraph on the left could be converted into a layout of edge and vertex gadgets as shown on the right. In this example, we use a grid layout; in general, we could use any layout consistent with the edge and vertex gadgets.

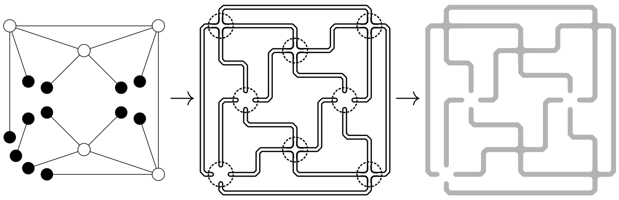

Figure 5 A choice of which vertices to break in the input multigraph (left) corresponds to a choice of local solutions at each of the breakable vertex gadgets, thereby yielding a candidate solution to the single-traversal instance (center). As a result, the shape of the interior of the candidate solution (right) is essentially the same as the shape of the residual multigraph after breaking vertices.

produced will satisfy all local constraints, but might still not satisfy the global (single cycle) constraint. Notice that the candidate solution is the boundary of the region "inside" the local solutions to the edge and vertex gadgets, and that this region ends up being the same shape as the multigraph obtained after breaking vertices. See Figure 5 for an example. The boundary of this region is a single cycle if and only if the region is connected and hole-free. Since the shape of this region is the same as the shape of the multigraph obtained after breaking vertices, this condition on the region's shape is equivalent to the condition that the residual multigraph must be connected and acyclic, or in other words, a tree. Thus, this construction yields a correct reduction, and in general this proof idea can be used to show NP-hardness of single-traversal problems.

Outline

In Section 2, we give an example of an NP-hardness proof following the above strategy. By reducing from TRVB, we give a simple proof that Hamiltonian Cycle in max-degree-3 square grid graphs is NP-hard (a result previously shown in [3]). We also use the same proof idea in manuscript [1] to show the novel result that Hamiltonian Cycle in hexagonal thin grid graphs is NP-hard.

In Section 3, we formally define the variants of TRVB under consideration. In the full version of this paper, we prove membership in NP and provide the obvious reductions between the variants.

Sections 4–7 address our NP-hardness results. In Section 4, we reduce from an NP-hard problem to show that Planar TRVB with only degree-k breakable vertices and unbreakable degree-4 vertices is NP-hard for any $k \geq 4$. All the other hardness results in this paper are derived directly or indirectly from this one. In Section 5, we prove the NP-completeness of the variants of TRVB and of Planar TRVB in which breakable vertices of some degree $k \geq 4$ are allowed. Similarly, we show in Section 6 that Graph TRVB is also NP-complete in the presence of breakable vertices of degree $k \geq 4$. Finally, in Section 7, we show that Planar Graph TRVB is NP-complete provided (1) breakable vertices of some degree $k \geq 4$ are allowed and (2) either breakable vertices of degree $b \leq 5$ or unbreakable vertices of degree $u \leq 4$ are allowed.

Next, in Section 8, we proceed to one of our polynomial-time results: that a variant of TRVB is solvable in polynomial time whenever the multigraph is restricted to be a planar graph, the breakable vertices are restricted to have degree at least 6, and the unbreakable vertices are restricted to have degree at least 5. In such a graph, it is impossible to break a set of breakable vertices and get a tree. As a result, variants of TRVB satisfying these restrictions are always solvable with a trivial polynomial time algorithm.

In Section 9, we establish a connection between TRVB and the Hypergraph Spanning Tree problem (given a hypergraph, decide whether it has a spanning tree). Namely, Hypergraph Spanning Tree on a hypergraph is equivalent to TRVB on the corresponding incidence graph with edge nodes marked breakable and vertex nodes marked unbreakable. This equivalence allows us to construct a reduction from TRVB to Hypergraph Spanning Tree: given a TRVB instance, we can first convert that instance into a bipartite TRVB instance (by inserting unbreakable vertices between adjacent breakable vertices and merging adjacent unbreakable vertices) and then construct the hypergraph whose incidence graph is the bipartite TRVB instance.

This connection allows us to obtain results about both TRVB and Hypergraph Spanning Tree. By leveraging known results about Hypergraph Spanning Tree (see [2]), we prove that TRVB is polynomial-time solvable when all breakable vertices have small degrees ($B \subseteq \{1, 2, 3\}$). This final result completes our classification of the variants of TRVB. We also apply the hardness results from this paper to obtain new results about Hypergraph Spanning Tree: namely, Hypergraph Spanning Tree is NP-complete in k-uniform 2-regular hypergraphs for any $k \geq 4$, even when the incidence graph of the hypergraph is planar. This improves the previously known result that Hypergraph Spanning Tree is NP-complete in k-uniform hypergraphs for any $k \geq 4$ (see [5]).

2 Example of how to use TRVB: Hamiltonicity in max-degree-3 square grid graphs

In this section, we show one example of using TRVB to prove hardness of a single-traversal problem. Namely, the result that Hamiltonian Cycle in max-degree-3 square grid graphs is NP-hard [3] can be reproduced with the following much simpler reduction.

The reduction is from the variant of TRVB in which the input multigraph is restricted to be planar and to have only degree-4 breakable vertices, which is shown NP-complete in Section 5. Given a planar multigraph G with only degree-4 breakable vertices, we output a max-degree-3 square grid graph by appropriately placing breakable degree-4 vertex gadgets (shown in Figure 7) and routing edge gadgets (shown in Figure 6) to connect them. The appropriate placement of gadgets can be accomplished in polynomial time by the results from [6]. Each edge gadget consists of two parallel paths of edges a distance of two apart, and as shown in the figure, these paths can turn, allowing the edge to be routed as necessary (without parity constraints). Each breakable degree-4 vertex gadget joins four edge gadgets in the configuration shown. Note that, as desired, the maximum degree of any vertex in the resulting grid graph is 3.

Consider any candidate set of edges C that could be a Hamiltonian cycle in the resulting grid graph. In order for C to be a Hamiltonian cycle, C must satisfy both the local constraint that every vertex is incident to exactly two edges in C and the global constraint that C is a cycle (rather than a set of disjoint cycles). It is easy to see that, in order to satisfy the local constraint, every edge in every edge gadget must be in C. Similarly, there are only two possibilities within each breakable degree-4 vertex gadget which satisfy the local constraint. These possibilities are shown in Figure 8.

We can identify the choice of local solution at each breakable degree-4 vertex gadget with the choice of whether to break the corresponding vertex. Under this bijection, every candidate solution C satisfying local constraints corresponds with a possible multigraph G' formed from G by breaking vertices. The key insight is that the shape of the region R inside C is exactly the shape of G'. This is shown for an example graph-piece in Figure 9.

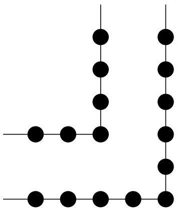

Figure 6 An edge gadget consisting of two parallel paths a distance of 2 apart.

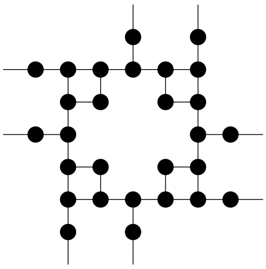

Figure 7 A degree-4 breakable vertex gadget.

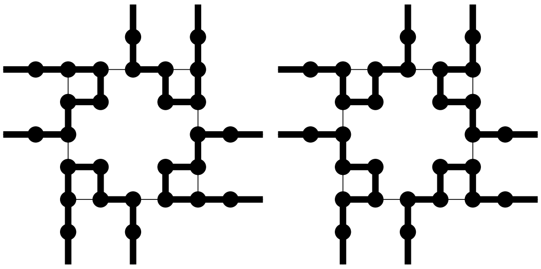

Figure 8 The two possible solutions to the vertex gadget from Figure 7 that satisfy the local constraints imposed by the Hamiltonian Cycle problem (broken on the left and unbroken on the right).

The boundary of R, also known as C, is exactly one cycle if and only if R is connected and hole-free. Since the shape of region R is the same as the shape of multigraph G', this corresponds to the condition that G' is connected and acyclic, or in other words that G' is a tree. Thus, there exists a candidate solution C to the Hamiltonian Cycle instance (satisfying the local constraints) that is an actual solution (also satisfying the global constraints) if and only if G is a "yes" instance of TRVB. Therefore, Hamiltonian Cycle in max-degree-3 square grid graphs is NP-hard.

3 Problem variants

In this section, we will formally define the variants of TRVB under consideration. In the full version of the paper, we also prove some basic results about these variants.

To begin, we formally define the TRVB problem. The multigraph operation of *breaking* vertex v in undirected multigraph G results in a new multigraph G' by removing v, adding a number of new vertices equal to the degree of v in G, and connecting these new vertices to the neighbors of v in G in a one-to-one manner (as shown in Figure 1 in Section 1). Using this definition, we pose the TRVB problem:

▶ **Problem 1.** *The* Tree-Residue Vertex-Breaking Problem (TRVB) *takes as input a multigraph G whose vertices are partitioned into two sets V_B and V_U (called the* breakable *and* unbreakable *vertices respectively), and asks to decide whether there exists a set $S \subseteq V_B$ such that after breaking every vertex of S in G, the resulting multigraph is a tree.*

In order to avoid trivial cases, we consider only input graphs that have no degree-0 vertices.

Next, suppose B and U are both sets of positive integers. Then we can constrain the breakable vertices of the input to have degrees in B and constrain the unbreakable vertices of the input to have degrees in U. The resulting constrained version of the problem is defined below:

▶ **Definition 2.** The *(B, U)-variant of the TRVB problem*, denoted (B, U)-TRVB, is the special case of TRVB where the input multigraph is restricted so that every breakable vertex in G has degree in B and every unbreakable vertex in G has degree in U.

Throughout this paper we consider only sets B and U for which membership can be computed in pseudopolynomial time (i.e., membership of n in B or U can be computed in time polynomial in n). As a result, verifying that the vertex degrees of a given multigraph are allowed can be done in polynomial time.

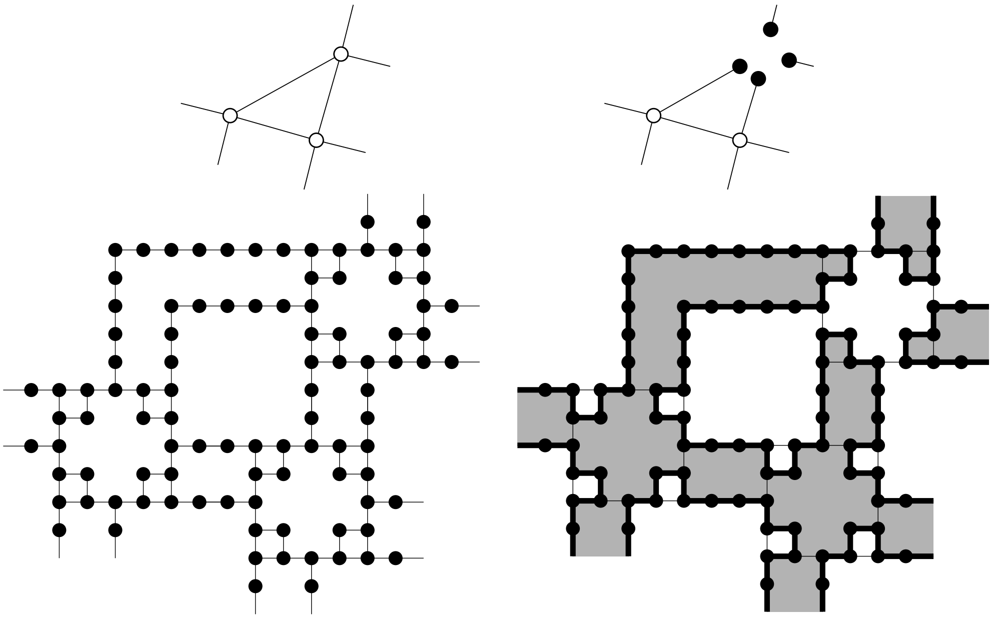

Figure 9 Given a multigraph including the piece shown in the top left, the output grid graph might include the section shown in the bottom left (depending on graph layout). If the top vertex in this piece of the multigraph is broken, resulting in the piece of multigraph G' shown in the top right, then the resulting candidate solution C (shown in bold) in the bottom right contains region R (shown in grey) whose shape resembles the shape of G'.

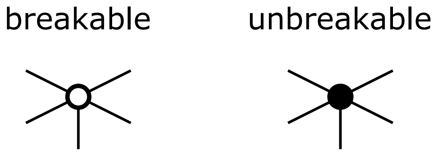

Figure 10 Depiction of vertex types in this paper.

We can also define three further variants of the problem depending on whether G is constrained to be planar, a (simple) graph, or both: the *Planar* (B, U)*-variant of the TRVB problem* (denoted Planar (B, U)-TRVB), the *Graph* (B, U)*-variant of the TRVB* (denoted Graph (B, U)-TRVB), and the *Planar Graph* (B, U)*-variant of the TRVB problem* (denoted Planar Graph (B, U)-TRVB).

3.1 Diagram conventions

Throughout this paper, when drawing diagrams, we will use filled circles to represent unbreakable vertices and unfilled circles to represent breakable vertices. See Figure 10.

4 Planar $(\{k\}, \{4\})$-TRVB is NP-hard for any $k \geq 4$

The overall goal of this section is to prove NP-hardness for several variants of TRVB. In particular, we will introduce an NP-hard variant of the Hamiltonicity problem in Section 4.1 and then reduce from this problem to Planar $(\{k\}, \{4\})$-TRVB for any $k \geq 4$ in Section 4.2. This is the only reduction from an external problem in this paper. All further hardness results will be derived from this one via reductions between different TRVB variants.

4.1 Planar Hamiltonicity in Directed Graphs with all in- and out-degrees 2 is NP-hard

The following problem was shown NP-complete in [4]:

▶ **Problem 3.** *The* Planar Max-Degree-3 Hamiltonicity Problem *asks for a given planar directed graph whose vertices each have total degree at most* 3 *whether the graph is Hamiltonian (has a Hamiltonian cycle).*

For the sake of simplicity we will assume that every vertex in an input instance of the Planar Max-Degree-3 Hamiltonicity problem has both in- and out-degree at least 1 (and therefore at most 2). This is because the existence of a vertex with in- or out-degree 0 in a graph immediately implies that there is no Hamiltonian cycle in that graph.

As it turns out, this problem is not quite what we need for our reduction, so below we introduce several new definitions and define a new variant of the Hamiltonicity problem:

▶ **Definition 4.** Call a vertex $v \in G$ *alternating* for a given planar embedding of a planar directed graph G if, when going around the vertex, the edges switch from inward to outward oriented more than once. Otherwise, call the vertex *non-alternating.* A non-alternating vertex has all its inward oriented edges in one contiguous section and all its outward oriented edges in another; an alternating vertex on the other hand alternates between inward and outward sections more times.

We call a planar embedding of planar directed graph G a *planar non-alternating embedding* if every vertex is non-alternating under that embedding. If G has a planar non-alternating embedding we say that G is a *planar non-alternating graph.*

▶ **Problem 5.** *The* Planar Non-Alternating Indegree-2 Outdegree-2 Hamiltonicity Problem *asks, for a given planar non-alternating directed graph whose vertices each have in- and out-degree exactly* 2*, whether the graph is Hamiltonian*

In the full version of this paper we prove that this problem is NP-hard by reducing from the Planar Max-Degree-3 Hamiltonicity Problem:

▶ **Theorem 6.** *The Planar Non-Alternating Indegree-*2 *Outdegree-*2 *Hamiltonicity Problem is NP-hard.*

4.2 Reduction to Planar $(\{k\}, \{4\})$-TRVB for any $k \geq 4$

Consider the following algorithm R_k:

▶ **Definition 7.** For $k \geq 4$, algorithm R_k takes as input a planar non-alternating graph G whose vertex in- and out-degrees all equal 2, and outputs an instance M' of Planar $(\{k\}, \{4\})$-TRVB.

To begin, we construct a labeled undirected multigraph M as follows; refer to Figure 11.

First we build all the vertices (and vertex labels) of M. For each vertex in G, we include an unbreakable vertex in M and for each edge in G we include a breakable vertex in M. If v is a vertex or e is an edge of G, we define $m(v)$ and $m(e)$ to be the corresponding vertices in M.

Next we add all the edges of M. Fix vertex v in G. Let (u_1, v) and (u_2, v) be the edges into v and let (v, w_1) and (v, w_2) be the edges out of v. Then add the following edges to M:

- Add an edge from $m(v)$ to each of $m((u_1, v))$, $m((u_2, v))$, $m((v, w_1))$, and $m((v, w_2))$.
- Add an edge from $m((v, w_1))$ to $m((v, w_2))$.
- Add $k - 3$ edges from $m((u_1, v))$ to $m((u_2, v))$.

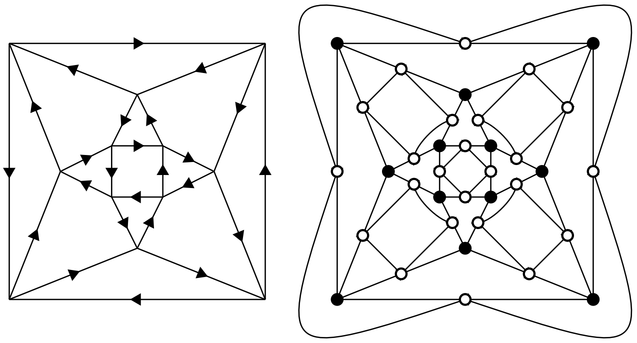

Figure 11 If the planar non-alternating directed graph on the left is G, and if $k = 4$, then we first produce multigraph M on the right. If $k > 4$, then the output M remains the same except some edges are duplicated.

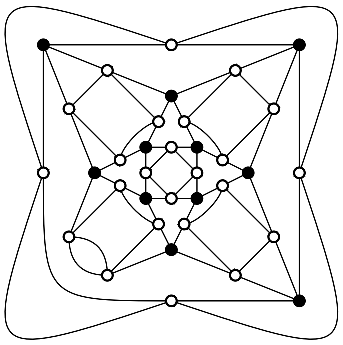

Figure 12 We modify M in the vicinity of one vertex $\hat{v}$ to get the output M' of our reduction. This figure shows one possible M' for the M in Figure 11, where $\hat{v}$ is chosen to be the bottom left vertex.

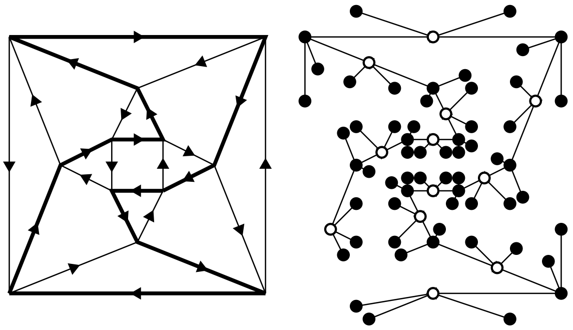

Figure 13 This figure shows a Hamiltonian cycle in example graph G from Figure 11 (left) and the corresponding solution of TRVB instance M' shown in Figure 12 (right).

Finally, pick any specific vertex $\hat{v}$ in G; refer to Figure 12. Let $(u_1, \hat{v})$ and $(u_2, \hat{v})$ be the edges into $\hat{v}$ and let $(\hat{v}, w_1)$ and $(\hat{v}, w_2)$ be the edges out of $\hat{v}$. We modify M by removing vertex $m(\hat{v})$ (and all incident edges), and adding the two edges $(m((u_1, \hat{v})), m((u_2, \hat{v})))$, and $(m((\hat{v}, w_1)), m((\hat{v}, w_2)))$. Call the resulting multigraph M' and return it as output of algorithm R_k.

We prove in the full version of this paper that algorithm R_k is a polynomial time reduction from the Planar Non-Alternating Indegree-2 Outdegree-2 Hamiltonicity Problem to Planar $(\{k\}, \{4\})$-TRVB. Figure 13 demonstrates the correspondence between a Hamiltonian Cycle in input G and a TRVB solution in output $R_k(G) = M'$. Thus we have the following:

▶ **Theorem 8.** *Planar $(\{k\}, \{4\})$-TRVB is NP-hard for any $k \geq 4$.*

5 Planar TRVB and TRVB are NP-complete with high-degree breakable vertices

▶ **Theorem 9.** *Planar (B, U)-TRVB is NP-complete if B contains any $k \geq 4$. Also (B, U)-TRVB is NP-complete if B contains any $k \geq 4$.*

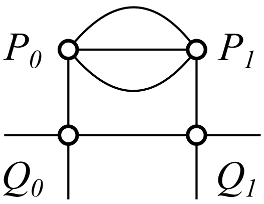

Figure 14 A gadget simulating an unbreakable degree-4 vertex using a planar arrangement of only breakable degree-4 vertices.

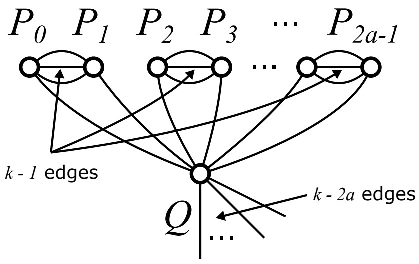

Figure 15 A gadget simulating an unbreakable degree-$(k-2a)$ vertex using only breakable degree-k vertices arranged in a planar manner. For $k > 4$, choosing a appropriately yields an unbreakable degree-3 or degree-4 gadget.

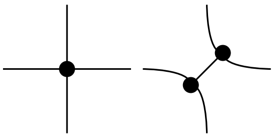

Figure 16 The degree-4 unbreakable vertex on the left can be simulated with two degree-3 unbreakable vertices as shown on the right while maintaining planarity.

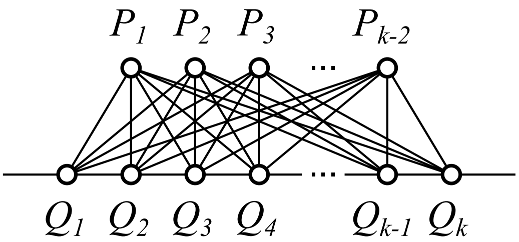
P_1 P_2 P_3 P_{k-2}
Q_1 Q_2 Q_3 Q_4 Q_{k-1} Q_k

Figure 17 A gadget simulating an unbreakable degree-2 vertex using only breakable degree-k vertices arranged without self loops or duplicated edges.

The basic idea for this theorem is to reduce from Planar $(\{k\}, \{4\})$-TRVB to Planar $(\{k\}, \emptyset)$-TRVB by creating a gadget which simules the behavior of an unbreakable degree-4 vertex using only breakable degree-k vertices. Figures 14, 15, and 16 sketch the construction of this gadget.

6 Graph TRVB is NP-complete with high-degree breakable vertices

▶ **Theorem 10.** *Graph (B, U)-TRVB is NP-complete if B contains any $k \geq 4$.*

The basic idea for this theorem is to reduce from (B, U)-TRVB by inserting a gadget into each edge which behaves like a degree-2 unbreakable vertices and which is built entirely out of breakable degree-k vertices. This converts the multigraph into a simple graph without affecting the answer of the TRVB instance and without adding any new values to B or U. Figure 17 sketches the construction of this gadget.

7 Planar Graph TRVB is NP-hard with both low-degree vertices and high-degree breakable vertices

▶ **Theorem 11.** *Planar Graph (B, U)-TRVB is NP-complete if (1) either $B \cap \{1, 2, 3, 4, 5\} \neq \emptyset$ or $U \cap \{1, 2, 3, 4\} \neq \emptyset$ and (2) there exists a $k \geq 4$ with $k \in B$.*

As in the previous section, the idea for this theorem is to use unbreakable degree-2 vertex gadgets to reduce from Planar (B, U)-TRVB, converting the input multigraph into a simple graph. We build such a gadget in one of several ways, depending on which vertex types are present. Figures 18–24 sketch the gadget construction for the various cases. See the full version of this paper for details.

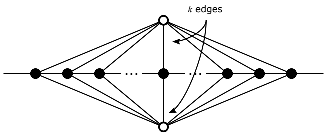

Figure 18 A gadget simulating an unbreakable degree-2 vertex using only breakable degree-k and unbreakable degree-4 vertices arranged in a planar manner without self loops or duplicate edges.

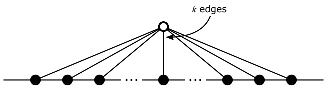

Figure 19 A gadget simulating an unbreakable degree-2 vertex using only breakable degree-k and unbreakable degree-3 vertices arranged in a planar manner without self loops or duplicate edges.

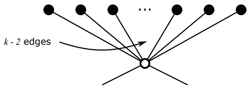

Figure 20 A gadget simulating an unbreakable degree-2 vertex using only breakable degree-k and unbreakable degree-1 vertices arranged in a planar manner without self loops or duplicate edges.

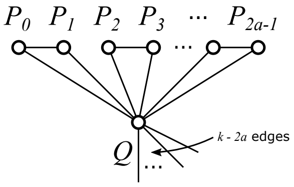

Figure 21 A gadget simulating an unbreakable degree-$(k - 2a)$ vertex using only breakable degree-k and degree-2 vertices arranged in a planar manner without self loops or duplicate edges.

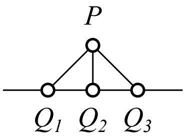

Figure 22 A gadget simulating an unbreakable degree-2 vertex using only breakable degree-3 vertices arranged in a planar manner without self loops or duplicate edges.

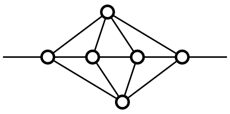

Figure 23 A gadget simulating an unbreakable degree-2 vertex using only breakable degree-4 vertices arranged in a planar manner without self loops or duplicate edges.

8 Planar Graph TRVB is polynomial-time solvable without small vertex degrees

The overall purpose of this section is to show that variants of Planar Graph TRVB which disallow all small vertex degrees are polynomial-time solvable because the answer is always "no." Consider for example the following theorem.

▶ **Theorem 12.** *If $b > 5$ for every $b \in B$ and $u > 5$ for every $u \in U$, then Planar Graph (B, U)-TRVB has no "yes" inputs. As a result, Planar Graph (B, U)-TRVB problem is polynomial-time solvable.*

Proof. The average degree of a vertex in a planar graph must be less than 6, so there are no planar graphs with all vertices of degree at least 6. Thus, if $b > 5$ for every $b \in B$ and $u > 5$ for every $u \in U$, then every instance of Planar Graph (B, U)-TRVB is a "no" instance. ◀

In fact, we will strengthen this theorem below to disallow "yes" instances even when degree-5 unbreakable vertices are present by using the particular properties of the TRVB problem. Note that this time, planar graph inputs which satisfy the degree constraints are possible, but any such graph will still yield a "no" answer to the Tree-Residue Vertex-Breaking problem.

We describe the proof idea in Section 8.1 with details available in the full version of the paper.

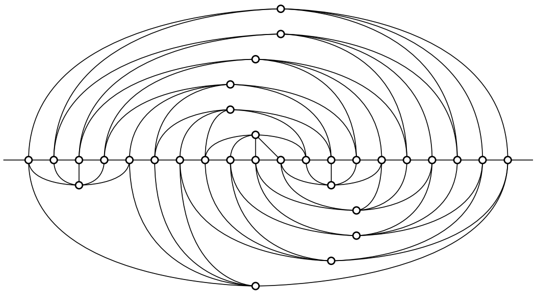

Figure 24 A gadget simulating an unbreakable degree-2 vertex using only breakable degree-5 vertices arranged in a planar manner without self loops or duplicate edges.

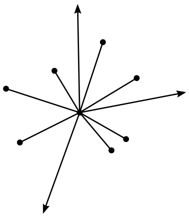

Figure 25 A degree-10 vertex with seven degree-1 neighbors (shown) and three other neighbors (not shown). The edges to the degree-1 neighbors form two bundles of size 2 and one bundle of size 3.

8.1 Proof idea

Consider the hypothetical situation in which we have a solution to the TRVB problem in a planar graph whose unbreakable vertices each have degree at least 5 and whose breakable vertices each have degree at least 6. The general idea of the proof is to show that this situation is impossible by assigning a scoring function (described below) to the possible states of the graph as vertices are broken. The score of the initial graph can easily be seen to be zero and assuming the TRVB instance has a solution, the score of the final tree can be shown to be positive. It is also the case, however, that if we break the vertices in the correct order, no vertex increases the score when broken, implying a contradiction.

Next, we introduce the scoring mechanism. Consider one vertex in the graph after some number of vertices have been broken. This vertex has several neighbors, some of which have degree 1. We can group the edges of this vertex that lead to degree-1 neighbors into “bundles” seperated by the edges leading to higher degree neighbors. For example, in Figure 25, the vertex shown has two bundles of size 2 and one bundle of size 3. Each bundle is given a score according to its size, and the score of the graph is equal to the cumulative score of all present bundles. In particular, if a bundle has a size of 1, then we assign the bundle a score of -1, and otherwise we assign the bundle a score of $n-1$ where n is the size of the bundle.

As it turns out, under this scoring mechanism, any tree all of whose non-leaves have degree at least 5 always has a positive score. In fact, it is easy to see that in our TRVB instance, if breaking some set of breakable vertices S results in a tree, then this degree constraint applies: the non-leaves are vertices from the original graph and therefore have degree at least 5. Thus, the score of the original graph is zero (since there are no bundles), and the score after all the vertices in S are broken is positive.

Next, we define a breaking order for the vertices of S. In short, we will break the vertices of S starting on the exterior of the graph and moving inward. More formally, we will repeatedly do the following step until all vertices in S have been broken. Consider the external face of the graph at the current stage of the breaking process. Since not every vertex in S has been broken, the graph is not yet a tree and the current external face is a cycle. Every cycle in the graph must contain a vertex from S (in order for the final graph to be a tree), so choose a vertex from S on the current external face and break that vertex next.

Breaking the vertices of S in this order has an interesting effect on the bundles in the graph: since every vertex from S is on the external face when it is broken, every degree-1 vertex ends up within the external face when it appears. Thus all bundles are within the external face of the graph at all times.

Consider the effect that breaking one vertex from S with degree $d \geq 6$ has on the score of the graph. Any vertex in S on the external face has exactly two edges which border this face. The remaining $d - 2$ edges must all leave the vertex into the interior of the graph. When the vertex is broken, each of these $d - 2$ edges becomes a new bundle (since the interior of the graph never has any bundles). Thus, breaking the vertex creates $d - 2$ new bundles of size 1, thereby decreasing the score of the graph by $d - 2$. On the other hand, the two edges which were on the external face are now each added to a bundle, thereby increasing the size of that bundle by one and increasing its score by at most two (in the case that the size was originally 1). Thus, the increase in the score of the graph due to these two edges is at most 4. In summary, breaking one vertex decreases the graph's score by $d - 2 \geq 4$ and increases the graph's score by at most 4. Thus, the total score of the graph does not increase.

Since the score of the graph does not increase with any step of the process, the final result should have at most the same score as the original graph. This contradicts the fact that the tree at the end of the process has positive score while the original graph has score zero. By contradiction, we conclude that S cannot exist, giving us our desired result.

▶ **Theorem 13.** *If $b > 5$ for every $b \in B$ and $u > 4$ for every $u \in U$, then Planar Graph (B, U)-TRVB can be solved in polynomial time.*

9 TRVB and the Hypergraph Spanning Tree problem

In the full version of this paper, we demonstrate the connection between the TRVB problem and the Hypergraph Spanning Tree problem.

In particular, we reduce from (B, U)-TRVB with $B \subseteq \{1, 2, 3\}$ to a version of the Hypergraph Spanning Tree problem in which the hypergraphs are restricted to have only edges with at most 3 endpoints. The Hypergraph Spanning Tree problem in such hypergraphs is known to be polynomial-time solvable (see [2]), so we can conclude the following:

▶ **Theorem 14.** *(B, U)-TRVB with $B \subseteq \{1, 2, 3\}$ is polynomial-time solvable.*

We also reduce from Planar $(\{k\}, \emptyset)$-TRVB to a version of the Hypergraph Spanning Tree problem in which the hypergraphs are restricted to be k-uniform and 2-regular and to have planar incidence graphs. Applying the fact that Planar $(\{k\}, \emptyset)$-TRVB is NP-hard for any $k \geq 4$, we immediately obtain the following:

▶ **Theorem 15.** *The Hypergraph Spanning Tree problem is NP-complete in k-uniform 2-regular hypergraphs for any $k \geq 4$, even when the incidence graph of the hypergraph is planar.*

References

1 Erik D. Demaine and Mikhail Rudoy. Hamiltonicity is hard in thin or polygonal grid graphs, but easy in thin polygonal grid graphs. arXiv:1706.10046, 2017. https://arxiv.org/abs/1706.10046.

2 László Lovász. Matroid matching and some applications. *J. Comb. Theory, Ser. B*, 28(2):208–236, 1980. doi:10.1016/0095-8956(80)90066-0.

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

4 Ján Plesník. The np-completeness of the hamiltonian cycle problem in planar digraphs with degree bound two. *Inf. Process. Lett.*, 8(4):199–201, 1979. doi:10.1016/0020-0190(79)90023-1.

5 Hans Jürgen Prömel and Angelika Steger. *The Steiner tree problem: a tour through graphs, algorithms, and complexity.* Springer Science & Business Media, 2002.

6 Markus W. Schäffter. Drawing graphs on rectangular grids. *Discrete Applied Mathematics*, 63(1):75–89, 1995. doi:10.1016/0166-218X(94)00020-E.