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

Erik D. Demaine∗  Mikhail Rudoy†

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 (disconnecting the edges at a degree-$k$ breakable vertex by replacing that vertex with $k$ degree-1 vertices)? We consider the special cases 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 $U$. We fully characterize these variants of TRVB as polynomially solvable or NP-complete. 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.

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 operation: disconnecting the edges at a degree-$k$ breakable vertex by replacing that vertex with $k$ degree-1 vertices (as shown in Figure 1).

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).

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

∗MIT Computer Science and Artificial Intelligence Laboratory, 32 Vassar St., Cambridge, MA 02139, USA, edemaine@mit.edu
†Google Inc, mrudoy@gmail.com Work completed at MIT Computer Science and Artificial Intelligence Laboratory.
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 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 degree $b \leq 5$ or unbreakable vertices of 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 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.

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. Notice that variants of the Hamiltonian Cycle problem fall under the category of single-traversal problems, but that not all problems in this category are necessarily variants of Hamiltonian Cycle (e.g. a single-traversal problem might allow the solution path/cycle to skip certain vertices entirely while still 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.

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 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.

Figure 2: The abstraction of a possible edge gadget containing two parallel paths (left), together with the local solution of that gadget (right). The bold paths are (forced to be) part of the traversal while the “inside” of the gadget is shown in grey.

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 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.

In Section 2, we give an example of an NP-hardness proof following this strategy. By reducing
Figure 3: The 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.

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.
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 [2]).

In Section 3, we formally define the variants of TRVB under consideration. We also prove membership in NP for all the variants and provide the obvious reductions between them.

Sections 4 and 5 address the polynomial-time results of this paper. In Section 4, we prove that, if every breakable vertex is restricted to have degree at most 3, then the TRVB variant in question is solvable in polynomial time. Next, in Section 5 we prove that if 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, then it is impossible to break a set of breakable vertices and get a tree. Therefore, in these cases, the TRVB variant is solvable in polynomial time.

Next we prove our NP-hardness results. In Section 6, we reduce from an NP-hard problem to show that Planar TRVB with only degree-4 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 7, 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 8 that Graph TRVB is also NP-complete in the presence of breakable vertices of degree \( k \geq 4 \). Finally, in Section 9 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.

All together, these results completely classify the variants of TRVB into NP-complete and polynomially solvable.

## 2. An 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 [2] 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 7. 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 6) and routing edge gadgets (shown in Figure 7) to connect them. The appropriate placement of gadgets can be accomplished in polynomial time by the results from [4]. 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 residual 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. 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, and prove some basic results about them.

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 3.1. 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. This means that the classification of a particular $(B,U)$-variant of the TRVB problem into P or NP-complete is a statement about the hardness of checking TRVB (while constrained by the other conditions) rather than a statement about the hardness of checking
membership in $B$ or $U$ for the degrees in the multigraph. In fact, all the results in this paper will also apply even in the cases that membership in $B$ or $U$ cannot be computed in pseudo-polynomial time if we consider the promise problems in which the given multigraph’s vertex degrees are guaranteed to comply with the sets $B$ and $U$.

We can also define three further variants of the problem depending on whether $G$ is constrained to be planar, a 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).

Note that since both being planar and being a graph are properties of a multigraph that can be verified in polynomial time, again the classification of these variants into P or NP-complete is a statement about the hardness of 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.

![breakable unbreakable](breakable.png)

Figure 10: An example diagram, showing the depictions of vertex types used in this paper.

### 3.2 Trivial reductions

As mentioned above, except for the constraint that the TRVB problem outputs “yes” on the given input, every other constraint in the definition of each of the above variants can be tested in polynomial time. Therefore, if for some two variants $X$ and $Y$ the non-TRVB conditions of $X$ are strictly stronger (more constraining) than the non-TRVB conditions of $Y$, then we can reduce from $X$ to $Y$ in polynomial time. In particular, we can convert an input $G$ for variant $X$ into an input $G'$ for $Y$ as follows:

First test all the non-TRVB conditions of variant $X$ on the input $G$. If any condition is not satisfied, then $X$ rejects $G$, so output any $G'$ rejected by $Y$. If all the non-TRVB conditions of variant $X$ are satisfied, then by assumption all the non-TRVB conditions of variant $Y$ on input $G$ are also satisfied. Therefore $G$ is a “yes” instance of both $X$ and $Y$ if and only if $G$ is a “yes” instance of TRVB. Therefore $X$ and $Y$ have the same answer on $G$, so outputting $G' = G$ completes the reduction.

Using the above reduction scheme, we conclude that:

**Lemma 3.2.** For any $(B,U)$, there are reductions

- from Planar $(B,U)$-TRVB to $(B,U)$-TRVB,
- from Graph $(B,U)$-TRVB to $(B,U)$-TRVB,
- from Planar Graph $(B,U)$-TRVB to Planar $(B,U)$-TRVB, and
- from Planar Graph $(B,U)$-TRVB to Graph $(B,U)$-TRVB.

For any $(B,U)$ and $(B',U')$ with $B \subseteq B'$ and $U \subseteq U'$, there are reductions
• from \((B,U)\)-TRVB to \((B',U')\)-TRVB,
• from Planar \((B,U)\)-TRVB to Planar \((B',U')\)-TRVB,
• from Graph \((B,U)\)-TRVB to Graph \((B',U')\)-TRVB, and
• from Planar Graph \((B,U)\)-TRVB to Planar Graph \((B',U')\)-TRVB.

3.3 Membership in NP

**Theorem 3.3.** The TRVB problem is in NP.

*Proof.* We describe a nondeterministic algorithm to solve TRVB: First nondeterministically guess a set of breakable vertices in \(G\). Break that set of vertices and accept if and only if the resulting multigraph is a tree.

This algorithm accepts an input \(G\) on at least one nondeterministic branch if and only if it is possible to break some of the breakable vertices so that the residual multigraph is a tree. In other words, this algorithm solves TRVB. Furthermore, the algorithm runs in polynomial time since both breaking vertices and checking whether a multigraph is a tree are polynomial-time operations. As desired, TRVB is in NP. \(\square\)

Another name for TRVB is \((N,N)\)-TRVB, so we can apply the reductions from Lemma 3.2 to conclude that:

**Corollary 3.4.** For any \((B,U)\), the \((B,U)\)-TRVB, Planar \((B,U)\)-TRVB, Graph \((B,U)\)-TRVB, and Planar Graph \((B,U)\)-TRVB are in NP.

4 TRVB is polynomial-time solvable without high-degree breakable vertices

The overall goal of this section is to show that the variants of TRVB without breakable vertices of degree \(k > 3\) are polynomial-time solvable. In particular, \((B,U)\)-TRVB is polynomial-time solvable if \(B \subseteq \{1, 2, 3\}\) (and thus so are Planar \((B,U)\)-TRVB, Graph \((B,U)\)-TRVB, and Planar Graph \((B,U)\)-TRVB).

We begin by proving that \((\{3\}, \emptyset)\)-TRVB is polynomial-time solvable in Section 4.1 as a warmup. We prove this by reducing \((\{3\}, \emptyset)\)-TRVB to the Cubic Nonseparating Independent Set problem, which was shown to be polynomial-time solvable in \([5]\). This warmup is useful because it introduces the Cubic Nonseparating Independent Set problem, points out the connection between this problem and Tree-Residue Vertex-Breaking, and proves a few useful lemmas; the actual proof that \((\{3\}, \emptyset)\)-TRVB is polynomial-time solvable (the main part of Section 4.1), however, is purely a warmup and can be safely skipped.

Next, in Section 4.2 we introduce the Nonseparating Independent Tag Set problem and prove that it is polynomial-time solvable. The Nonseparating Independent Tag Set problem is a generalization of the Cubic Nonseparating Independent Set problem, and the proof that Nonseparating Independent Tag Set is polynomial-time solvable follows the same approach as was used in \([5]\) for the Cubic Nonseparating Independent Set problem.

Finally, in Section 4.3 we reduce from \((\{1, 2, 3\}, N)\)-TRVB to Nonseparating Independent Tag Set, thereby concluding via Lemma 3.2 that \((B,U)\)-TRVB, Planar \((B,U)\)-TRVB, Graph \((B,U)\)-TRVB, and Planar Graph \((B,U)\)-TRVB with \(B \subseteq \{1, 2, 3\}\) can be solved in polynomial time.
4.1 \((\{3\}, \emptyset)\)-TRVB is polynomial-time solvable

We show that \((\{3\}, \emptyset)\)-TRVB (the Tree-Residue Vertex-Breaking problem with all vertices breakable and degree-3) is polynomial-time solvable by reduction to the Cubic Nonseparating Independent Set problem, which was shown to be solvable in polynomial time in [5]. We define the problem and related terms below:

**Definition 4.1.** A set of vertices \(X\) of multigraph \(G\) is a separating set if the number of connected components of \(G - X\) is more than that of \(G\).

**Definition 4.2.** A set of vertices \(S\) in a multigraph is an independent set if no two vertices in the set are adjacent and is a nonseparating set if every \(X \subseteq S\) is not a separating set.

**Problem 2.** The Cubic Nonseparating Independent Set problem asks, for a given cubic multigraph \(G\) and a given number \(s\), whether there exists a nonseparating independent set \(S\) in \(G\) with \(|S| = s\).

The following lemma is an important connection between vertex breaking and nonseparating independent sets.

**Lemma 4.3.** Suppose that \(G\) is a connected multigraph and \(S\) is a subset of the vertices in \(G\). Let \(G'\) be the graph formed by breaking all the vertices of \(S\) in \(G\). Then \(G'\) is connected if and only if \(S\) is a nonseparating independent set.

**Proof.** First suppose that \(G'\) is connected.

Let \(A\) be any subset of \(S\). Let \(G'_{A}\) be the multigraph formed by breaking all the vertices of \(A\) in \(G\). We know that \(G'\) can be formed from \(G'_{A}\) by breaking the vertices in \(S - A\). Since the reverse operation of vertex breaking is a merging of several degree-1 vertices, we conclude that we can construct \(G'_{A}\) from \(G'\) by merging vertices. Then since merging vertices only increases connectivity and \(G'\) is connected, we can conclude that \(G'_{A}\) is connected as well. Let \(G''_{A}\) be the multigraph formed by removing the vertices of \(A\) from \(G\). Vertex breaking is equivalent to vertex removal followed by an addition of a new degree-1 vertex (also adding that vertex’s sole edge). Therefore, it is possible to transform \(G''_{A}\) into \(G'_{A}\) by inserting new degree-1 vertices. Alternatively, this means we can transform \(G'_{A}\) into \(G''_{A}\) by removing degree-1 vertices. Removing a degree-1 vertex never disconnects a multigraph, so since \(G'_{A}\) is connected, we conclude that \(G''_{A}\) is connected. We have shown that removing the vertices of \(A\) from \(G\) (which yields \(G''_{A}\)) does not disconnect the multigraph. Therefore \(A\) is not a separating set of \(G\). Since this is true for every \(A \subseteq S\), we conclude that by definition, \(S\) is a nonseparating set.

Furthermore, no two vertices in \(S\) can be adjacent because that would lead to a separation of the edge between them from the rest of the multigraph (in \(G'\)), so \(S\) is an independent set. Thus, we have shown that if \(G'\) is connected, then \(S\) is a nonseparating independent set.

On the other hand, suppose \(S\) is a nonseparating independent set. Let \(G''\) be the multigraph formed by removing the vertices of \(S\) from \(G\).

Let \(T\) be the set of vertices in \(G\) but not \(S\). The set of vertices of \(G''\) is exactly \(T\). The set of vertices of \(G'\) contains \(T\). The only vertices in \(G'\) that are not in \(T\) are the degree-1 vertices that are added to \(G\) during the vertex breaking operation. Note that by construction, if \(t_1\) and \(t_2\) are in \(T\), then the edge \((t_1, t_2)\) is either in all three graphs \(G\), \(G'\), and \(G''\), or in none of them. Since \(S\) is nonseparating, \(G''\) has the same number of components as \(G\); precisely one. Then since \(T\) is the vertex set of \(G''\) and \(G''\) is connected, there is a path in \(G''\) between any two vertices in \(T\) (using only vertices in \(T\)). Every such path must also exist in \(G'\), and so all of \(T\) is in the same connected component of \(G'\).
Consider any vertex $v'$ in $G'$ that is not in $T$. This vertex $v'$ was added to $G'$ while breaking some vertex $v \in S$. At the time of the breaking operation, some neighbor $u$ of $v$ instead became the sole neighbor of $v'$. Since $v \in S$ and $S$ is an independent set in $G$, we know that $u$, which is a neighbor of $v$ in $G$ cannot also be in $S$. Thus $u \not\in S$ and therefore $u \in T$. Then the edge $(u, v')$ in $G'$ connects $v'$ to the connected component of $G'$ containing all of $T$. Since this applies to every $v' \not\in T$, we can conclude that all of $G'$ is in the same connected component. In other words, we have shown that $G'$ is connected whenever $S$ is a nonseparating independent set.

As desired, we have shown both directions. We conclude that $G'$ is connected if and only if $S$ is a nonseparating independent set.

Next, we prove the main theorem of this section, which as previously mentioned will serve as a warmup for the further results.

**Theorem 4.4.** $\{(3), \emptyset\}$-TRVB is polynomial-time solvable.

**Proof.** Suppose that $G$ is a multigraph in which every vertex is breakable and degree-3. Then if $G$ contains $n$ vertices, we know that $G$ must contain $\frac{3n}{2}$ edges. Breaking a degree-3 vertex leaves the number of edges unchanged and increases the number of vertices by 2. Therefore if a solution to the Tree-Residue Vertex-Breaking problem on $G$ breaks $b$ vertices, the final multigraph will have $n + 2b$ vertices and $\frac{3n}{2}$ edges.

In order for a solution to the Tree-Residue Vertex-Breaking problem on $G$ to be valid, the resulting multigraph must be a tree. Therefore the number of vertices is one more than the number of edges, or in other words $(n + 2b) = (\frac{3n}{2}) + 1$. Thus $b = \frac{n+2}{4}$.

We see that any valid solution to the Tree-Residue Vertex-Breaking problem on $G$ will break exactly $\frac{n+2}{4}$ vertices.

In combination with Lemma 4.3, this tells us that any valid solution to the Tree-Residue Vertex-Breaking problem on $G$ breaks a set of vertices $S$ where $S$ is a nonseparating independent set and $|S| = \frac{n+2}{4}$.

On the other hand, suppose $S$ is a nonseparating independent set of size $\frac{n+2}{4}$. By Lemma 4.3, the multigraph $G'$ formed by breaking the vertices of $S$ in $G$ is connected. Since every vertex has degree 3, the breaking of each element of $S$ in $G$ increases the number of vertices by 2 and leaves the number of edges the same. Thus the number of vertices in $G'$ is $n + 2(\frac{n+2}{4}) = \frac{3n}{2} + 1$ and the number of edges in $G'$ is $\frac{3n}{2}$. Since $G'$ is a connected multigraph which has one more vertex than it has edges, we can conclude that $G'$ is a tree. In other words, breaking the vertices of $S$ is a valid solution to the Tree-Residue Vertex-Breaking problem on $G$.

We conclude that the nonseparating independent sets of $G$ of size $\frac{n+2}{4}$ are exactly the sets of broken vertices from the valid solutions of the Tree-Residue Vertex-Breaking problem on $G$. Since the Cubic Nonseparating Independent Set problem can be solved in polynomial time, this allows us to also solve $\{(3), \emptyset\}$-TRVB in polynomial time.

### 4.2 Nonseparating Independent Tag Set is polynomial-time solvable

The purpose of this section is to introduce the following problem and prove that it is polynomial-time solvable:

**Problem 3.** Define a tagged graph to be a graph together with a partial labeling of the vertices such that (1) every vertex is labeled with either zero or one tags and (2) every tag either labels one degree-3 vertex or labels two degree-2 vertices.
The Nonseparating Independent Tag Set problem asks, for a given tagged graph $G$ and a given number $s$, whether there exists a set $S$ of tags such that $|S| = s$ and such that $S'$, the set of vertices in $G$ which are labeled with tags in $S$, is a nonseparating independent set.

Notice that on inputs that are cubic graphs whose vertices are each uniquely labeled, a solution to the Nonseparating Independent Tag Set problem corresponds exactly to a nonseparating independent set in that graph. In other words, the Nonseparating Independent Tag Set problem is a generalization of the Cubic Nonseparating Independent Set problem. The Cubic Nonseparating Independent Set problem was shown to be polynomial-time solvable in [5] via a reduction to the Matching problem for 2-polymatroids (introduced later in this section), which itself was shown to be polynomial-time solvable in [1]. We will expand upon this result, providing a very similar reduction from the more general Nonseparating Independent Tag Set problem to the Matching problem for 2-polymatroids.

To define the Matching problem for 2-polymatroids, we need to introduce several new terms.

**Definition 4.5.** A polymatroid is a pair $(X, f)$ where $X$ is a finite set and $f$ is a function defined on subsets of $X$ such that

1. $f(\emptyset) = 0$,
2. $f(A_1) \leq f(A_2)$ if $A_1 \subseteq A_2 \subseteq X$, and
3. $f(A_1 \cup A_2) + f(A_1 \cap A_2) \leq f(A_1) + f(A_2)$ for any $A_1, A_2 \subseteq X$.

A $k$-polymatroid is a polymatroid $(X, f)$ satisfying the additional condition that $f(\{x\}) \leq k$ for all $x \in X$. A matroid is a 1-polymatroid.

**Definition 4.6.** A matching of a 2-polymatroid $(X, f)$ is a set $Y \subseteq X$ such that $f(Y) = 2|Y|$.

At this point, we are almost ready to define the Matching problem for 2-polymatroids. The issue is that polymatroids in general are computationally intractible to deal with; just the description of a polymatroid $(X, f)$ has length exponential in $|X|$. This is not the case for a restricted subset of all polymatroids: “linearly represented” polymatroids.

“Linearly represented” polymatroids are defined in terms of linearly represented matroids below, where we treat linearly represented matroids as a black box whose definition will not be given in this paper. For the definition of and other information on linearly represented matroids, see [6].

**Definition 4.7.** A linearly represented polymatroid is a polymatroid with a “linear representation.” A linear representation of polymatroid $(X, f)$ is a linearly represented matroid $(E, r)$ and an assignment of subsets $E_x$ to elements $x$ of $X$ such that, for any $Y \subseteq X$,

$$f(Y) = r \left( \bigcup_{x \in Y} E_x \right).$$

With that done, we can finally define the Matching problem for 2-polymatroids:

**Problem 4.** The Matching problem for linearly represented 2-polymatroids asks, for a given linear representation of 2-polymatroid $(X, f)$ and a given number $s$, whether it is possible to find a matching of $(X, f)$ of size at least $s$.

We wish to reduce from the Nonseparating Independent Tag Set problem to the Matching problem for 2-polymatroids, so we need some way to connect graphs and polymatroids. This is done through the following definition.
Definition 4.8. Define the circuit rank $\mu(G)$ of graph $G$ to be $m - n + c$ where $m,$ $n,$ and $c$ are the edge, vertex, and connected-component counts in $G$.

Suppose $G$ is a graph. If $V$ is the set of vertices of $G,$ then we define $P(G)$ to equal $(V, f)$ where $f$ is a function defined on subsets of $V$ such that for $V' \subseteq V$ we define $f(V') = \mu(G) - \mu(G - V').$

Suppose next that $G$ is a tagged graph and $T$ is the set of tags in $G.$ Let $(V, f) = P(G),$ and for $t \in T$ let $e(t)$ be the set of vertices in $V$ labeled with tag $t.$ Then we define $\hat{P}(G) = (T, f')$ where $f'$ is a function defined on subsets of $T$ such that, for any $T' \subseteq T,$

$$f'(T') = f\left(\bigcup_{t \in T'} e(t)\right).$$

Then the reduction we are looking for takes as input a tagged graph $G$ and a number $s$ (an instance of Nonseparating Independent Tag Set) and outputs a linear representation of $\hat{P}(G)$ and $s$ (an instance of the Matching problem for 2-polymatroids).

Later in this section, we will prove two major lemmas, stated below, which will allow us to show that this reduction is both correct and computable in polynomial time:

**Lemma 4.9.** For any tagged graph $G,$ $\hat{P}(G)$ is a linearly representable 2-polymatroid whose linear representation can be identified in polynomial time.

**Lemma 4.10.** $T'$ is a matching of $\hat{P}(G) = (T, f')$ if and only if the set of vertices $S$ labeled with tags in $T'$ is a nonseparating independent set.

**Theorem 4.11.** The Nonseparating Independent Tag Set problem is polynomial-time solvable.

**Proof.** We can immediately conclude from Lemma 4.9 that the outputs of the reduction (a linear representation of $\hat{P}(G)$ and $s$) can be computed from the inputs ($G$ and $s$) in polynomial time. Lemma 4.9 also tells us that $\hat{P}(G)$ is a 2-polymatroid, which makes the output of the reduction a valid instance of the Matching problem for 2-polymatroids. In addition, we know from Lemma 4.10 that the sets of tags for which the corresponding sets of vertices are nonseparating independent sets are exactly the matchings in $\hat{P}(G).$ Thus, $\hat{P}(G)$ has a matching of size $s$ if and only if $G$ has a set of tags of size $s$ whose corresponding set of vertices is a nonseparating independent set. In other words, there exists a solution to the Nonseparating Independent Tag Set instance specified by $G$ and $s$ if and only if there exists a solution to the Matching for 2-polymatroids instance specified by a linear representation of $\hat{P}(G)$ and $s.$

Thus, the reduction is correct and can be run in polynomial time, so we can conclude that as desired, the Nonseparating Independent Tag Set problem is polynomial-time solvable.

All that is left is to prove the two lemmas above.

We begin with Lemma 4.9. It was shown in [5] that for any graph $G,$ $P(G)$ is a polymatroid. The paper then proves that a linear representation for $P(G)$ can be found in polynomial time. In order to make use of this fact, we will prove the following lemma:

**Lemma 4.12.** Suppose $(X, f)$ is a polymatroid, $(L, h)$ is a linearly represented polymatroid, and each element $x$ of $X$ is assigned a subset $L_x \subseteq L$ such that for any $Y \subseteq X,$

$$f(Y) = h\left(\bigcup_{x \in Y} L_x\right).$$

Then a linear representation for $(X, f)$ can be computed from a linear representation of $(L, h)$ in polynomial time.
Proof. Let \((E, r)\) and \(\{E_l\}_{l \in L}\) be a linear representation of \((L, h)\).

For each \(x \in X\) define \(E'_x\) to be \(\bigcup_{l \in L_x} E_l\). Then \((E, r)\) and \(\{E'_x\}_{x \in X}\) is the linear representation of \((X, f)\) whose existence is claimed by this lemma. We can certainly compute \((E, r)\) and \(\{E'_x\}_{x \in X}\) in polynomial time from \((E, r)\) and \(\{E_l\}_{l \in L}\), so provided this really is a linear representation, we will have shown our desired result.

In order for \((E, r)\) and \(\{E_l\}_{l \in L}\) to be the linear representation of \((L, h)\), the object \((E, r)\) must be a linearly representable matroid. And since \(E_l \subseteq E\) for all \(l \in L\), we have that \(E'_x = \bigcup_{l \in L_x} E_l \subseteq E\) for all \(x \in X\). Thus in order to show that \((E, r)\) and \(\{E'_x\}_{x \in X}\) is a linear representation of \((X, f)\), all that is left is to show that for any set \(Y \subseteq X\), it is the case that

\[
f(Y) = r\left(\bigcup_{x \in Y} E'_x\right).
\]

Define \(L_Y = \bigcup_{x \in Y} L_x\). We can apply the definition of \(E'_x\) to see that

\[
r\left(\bigcup_{x \in Y} E'_x\right) = r\left(\bigcup_{x \in Y} \bigcup_{l \in L_x} E_l\right).
\]

Then substituting in the definition of \(L_Y\), we see that

\[
r\left(\bigcup_{x \in Y} \bigcup_{l \in L_x} E_l\right) = r\left(\bigcup_{l \in L_Y} E_l\right).
\]

Next, we apply the definition of a linear representation to linear representation \((E, r)\) and \(\{E_l\}_{l \in L}\) of \((L, h)\) to get

\[
r\left(\bigcup_{l \in L_Y} E_l\right) = h(L_Y).
\]

But by definition of \(L_Y\),

\[
h(L_Y) = h\left(\bigcup_{x \in Y} L_x\right).
\]

Finally, applying the assumption given in this lemma, we see that

\[
h\left(\bigcup_{x \in Y} L_x\right) = f(Y).
\]

Putting this all together, we have that \(f(Y) = r\left(\bigcup_{x \in Y} E'_x\right)\) as desired. \(\square\)

With this lemma in place, we proceed to show that \(\hat{P}(G)\) is a linearly representable polymatroid whose linear representation can be identified in polynomial time.

**Lemma 4.13.** For any tagged graph \(G\), \(\hat{P}(G)\) is a linearly representable polymatroid whose linear representation can be identified in polynomial time.
Proof. Suppose that \( \hat{P}(G) = (T, f') \) and \( P(G) = (V, f) \).

We first verify that \((T, f')\) is a polymatroid.

Consider \( f'(\emptyset) \). This value is defined to be \( f(\bigcup_{t \in \emptyset} e(t)) = f(\emptyset) \), so since \( P(G) = (V, f) \) is a polymatroid, we have that \( f(\emptyset) = 0 \). Thus \( f'(\emptyset) = 0 \).

Consider any \( A_1 \subseteq A_2 \subseteq T \). We know that \( f'(A_1) = f\left(\bigcup_{t \in A_1} e(t)\right) \) and that \( f'(A_2) = f\left(\bigcup_{t \in A_2} e(t)\right) \). Then if we let \( B_1 = \bigcup_{t \in A_1} e(t) \) and \( B_2 = \bigcup_{t \in A_2} e(t) \), we see that \( B_1 \subseteq B_2 \subseteq V \), and therefore (since \( P(G) = (V, f) \) is a polymatroid) \( f(B_1) \leq f(B_2) \). But then \( f'(A_1) = f(B_1) \leq f(B_2) = f'(A_2) \), so \( f'(A_1) \leq f'(A_2) \).

Consider any \( A_1, A_2 \subseteq T \). Define \( B_1 = \bigcup_{t \in A_1} e(t) \) and \( B_2 = \bigcup_{t \in A_2} e(t) \). Then we have

\[
f'(A_1 \cup A_2) = f\left(\bigcup_{t \in A_1 \cup A_2} e(t)\right) = f\left(\bigcup_{t \in A_1} e(t) \cup \bigcup_{t \in A_2} e(t)\right) = f(B_1 \cup B_2),
\]

\[
f'(A_1) = f\left(\bigcup_{t \in A_1} e(t)\right) = f(B_1),
\]

and

\[
f'(A_2) = f\left(\bigcup_{t \in A_2} e(t)\right) = f(B_2).
\]

Using the fact that the sets \( e(t) \) are disjoint for different \( t \), we can also show that

\[
f'(A_1 \cap A_2) = f\left(\bigcup_{t \in A_1 \cap A_2} e(t)\right) = f\left(\bigcup_{t \in A_1} e(t) \cap \bigcup_{t \in A_2} e(t)\right) = f(B_1 \cap B_2).
\]

Then the statement \( f(B_1 \cup B_2) + f(B_1 \cap B_2) \leq f(B_1) + f(B_2) \) (true because \( P(G) = (V, f) \) is a polymatroid) can be rewritten as \( f'(A_1 \cup A_2) + f'(A_1 \cap A_2) \leq f'(A_1) + f'(A_2) \).

As we see, \((T, f')\) satisfies all the conditions in the definition of a polymatroid and therefore is one.

Next, notice that if \( T' \subseteq T \), then \( f'(T') = f\left(\bigcup_{t \in T'} e(t)\right) \). Because of this property, we can apply Lemma 4.12 it is possible to compute a linear representation for \( \hat{P}(G) = (T, f') \) from a linear representation of \( P(G) = (V, f) \). Since a linear representation for \( P(G) \) can be computed in polynomial time, this means that a linear representation for \( \hat{P}(G) \) can be computed in polynomial time.

We have shown that, for a tagged graph \( G \), the object \( \hat{P}(G) \) is a polymatroid whose linear representation can be computed in polynomial time. Consider Lemma 4.9 reproduced below. All that we have left to show in order to prove Lemma 4.9 is that \( \hat{P}(G) \) is actually a 2-polymatroid.

**Lemma 4.9.** For any tagged graph \( G \), \( \hat{P}(G) \) is a linearly representable 2-polymatroid whose linear representation can be identified in polynomial time.

**Proof.** Suppose that \( \hat{P}(G) = (T, f') \) and \( P(G) = (V, f) \). We must show that, for any \( t \in T \), \( f'(\{t\}) \leq 2 \).

There are two cases. Either \( t \) is a tag labeling two degree-2 nodes or \( t \) is a tag labeling one degree-3 node. If \( t \) is a tag labeling two degree-2 nodes, then \( e(t) = \{v_1, v_2\} \) where \( v_1, v_2 \in V \) have
degree 2. Then \( f'(\{t\}) = f(\{v_1, v_2\}) \leq f(\{v_1\}) + f(\{v_2\}) \). If \( t \) is a tag labeling one degree-3 node, then \( e(t) = \{v\} \) where \( v \in V \) has degree 3. Then \( f'(\{t\}) = f(\{v\}) \).

Note that for any vertex \( v \), \( f(\{v\}) = \mu(G) - \mu(G - \{v\}) \). If we define \( \Delta m, \Delta n, \) and \( \Delta c \) to be the increases in edge, vertex, and connected component counts due to adding vertex \( v \) into graph \( G - \{v\} \), then \( f(\{v\}) = \Delta m - \Delta n + \Delta c \). Note that \( \Delta m \) is the degree of \( v \) in \( G \), \( \Delta n = 1 \) and \( \Delta c \leq 0 \) (since adding a vertex that is connected to something else cannot increase the number of connected components). Therefore we see that for any vertex \( v \), \( f(\{v\}) \) is bounded above by the degree of \( v \) minus one.

Thus in the first case \( f'(\{t\}) = f(\{v_1, v_2\}) \leq f(\{v_1\}) + f(\{v_2\}) \leq (2 - 1) + (2 - 1) = 2 \) and in the second case \( f'(\{t\}) = f(\{v\}) \leq (3 - 1) = 2 \). In both cases, we have concluded that \( f'(\{t\}) \leq 2 \), so we can conclude that \( \hat{P}(G) \) is a 2-polymatroid.

With that done, we can proceed to the last remaining part of the proof: the proof of Lemma 4.10. Consider first the following lemma connecting \( \hat{P}(G) \) and \( G \):

**Lemma 4.14.** Define \( d_G(v) \) to be the degree of vertex \( v \) in graph \( G \). Suppose for some graph \( G \) we have \( P(G) = (V, f) \). Then for any set \( X \subseteq V \), \( f(X) = \sum_{v \in X} (d_G(v) - 1) \) if and only if \( X \) is a nonseparating independent set in \( G \).

**Proof.** We will first prove that if \( X \) is a nonseparating independent set of \( G \), then \( f(X) = \sum_{v \in X} (d_G(v) - 1) \).

We proceed by induction on the size of \( X \). Clearly, if \( X \) is a nonseparating independent set and \( |X| = 0 \), then \( f(X) = 0 = \sum_{v \in X} (d_G(v) - 1) \).

Next suppose that for any nonseparating independent set \( Y \) of size at most \( i - 1 \), we have that \( f(Y) = \sum_{v \in Y} (d_G(v) - 1) \) and let \( X \) be a nonseparating independent set of size \( i \). Choose some vertex \( x \in X \). The set \( X - \{x\} \) is a nonseparating set of size \( i - 1 \), so the inductive hypothesis applies and \( f(X - \{x\}) = \sum_{v \in X - \{x\}} (d_G(v) - 1) \). Then

\[
\begin{align*}
f(X) &= f(X) - f(X - \{x\}) \\
&= \mu(G) - \mu(G - X) = (\mu(G) - \mu(G - (X - \{x\}))) + (\mu(G - (X - \{x\})) - \mu(G - X)) \\
&= f(X - \{x\}) + (\mu(G - (X - \{x\})) - \mu(G - X)).
\end{align*}
\]

Since \( X \) is an independent set, \( x \) is not adjacent to any other member of \( X \) and so removing them from the graph does not affect the degree of \( x \). In other words \( d_G(x) = d_{G - (X - \{x\})}(x) \). Thus from graph \( G - X \) to graph \( G - (X - \{x\}) \), the number of vertices increases by one and the number of edges increases by \( d_{G}(x) \). Since \( X \) is a nonseparating set, the number of connected components in \( G \), \( G - (X - \{x\}) \), and \( G - X \) are all the same. Thus from graph \( G - X \) to graph \( G - (X - \{x\}) \), the number of connected components does not change. Putting these numbers together, we see that \( (\mu(G - (X - \{x\})) - \mu(G - X)) = d_{G}(x) - 1 \). Thus,

\[
f(X) = \left( \sum_{v \in X - \{x\}} (d_G(v) - 1) \right) + (d_{G}(x) - 1) = \sum_{v \in X} (d_G(v) - 1),
\]

as desired. By induction, we see that if \( X \) is a nonseparating independent set of any size, then \( f(X) = \sum_{v \in X} (d_G(v) - 1) \).

Next we prove the other direction: if \( X \) is not a nonseparating independent set, then \( f(X) \neq \sum_{v \in X} (d_G(v) - 1) \). We know that \( f(X) \) is defined as \( f(X) = \mu(G) - \mu(G - X) \). Then if we list the elements of \( X \) as \( x_1, x_2, ..., x_i \), we can rewrite \( f(X) \) as

\[
f(X) = \sum_{j=1}^{i} (\mu(G - \{x_1, ..., x_{j-1}\}) - \mu(G - \{x_1, ..., x_j\})).
\]
Notice that $\mu(G - \{x_1, \ldots, x_{j-1}\}) - \mu(G - \{x_1, \ldots, x_j\}) = d_{G - \{x_1, \ldots, x_{j-1}\}}(x_j) - 1 - \Delta c$ where $\Delta c$ is the decrease in the number of connected components between graph $G - \{x_1, \ldots, x_j\}$ and graph $G - \{x_1, \ldots, x_{j-1}\}$. We know that $d_{G - \{x_1, \ldots, x_{j-1}\}}(x_j) = d_G(x_j)$.

If $X$ is not an independent set, then some vertex $x_{j_1}$ will be adjacent to some other vertex $x_{j_2}$. Without loss of generality, let $j_1 < j_2$. Then in graph $G - \{x_1, \ldots, x_{j_2-1}\}$, vertex $x_{j_2}$ will have a smaller degree than in graph $G$ (since at least the neighbor $x_{j_1}$ is missing). Thus $d_{G - \{x_1, \ldots, x_{j_1-1}\}}(x_j) \neq d_G(x_j)$ and strict inequality holds at least in that case: $\mu(G - \{x_1, \ldots, x_{j-1}\}) - \mu(G - \{x_1, \ldots, x_j\}) < d_G(x_j) - 1$.

If $X$ is not a nonseparating set, then some minimal subset $A \subseteq X$ separates the graph. Suppose we order the $x_i$s such that $A = \{x_1, \ldots, x_j\}$. Then vertex $x_j$ will be a cut-vertex in graph $G - \{x_1, \ldots, x_{j-1}\}$. At that step, removing the vertex will change the number of connected components (i.e. $\Delta c \neq 0$). As a result, strict inequality will again hold: $\mu(G - \{x_1, \ldots, x_{j-1}\}) - \mu(G - \{x_1, \ldots, x_j\}) < d_G(x_j) - 1$.

Combining this upper bound over the summation, we can bound $f(x)$ from above by $\sum_{v \in X} (d_G(v) - 1)$. Notice that this is exactly our target value. And provided that $X$ is not a nonseparating independent set, at least one of the inequalities used to construct this bound will be strict. In that case, our target value cannot be achieved: $f(X) < \sum_{v \in X} (d_G(v) - 1)$. This concludes the proof that if $X$ is not a nonseparating independent set, then $f(X) \neq \sum_{v \in X} (d_G(v) - 1)$.

As desired, we have shown that $X$ is a nonseparating independent set if and only if $f(X) \neq \sum_{v \in X} (d_G(v) - 1)$.

With that done, we can prove Lemma 4.10 which is reproduced below:

**Lemma 4.10.** $T'$ is a matching of $\hat{P}(G) = (T, f')$ if and only if the set of vertices $S$ labeled with tags in $T'$ is a nonseparating independent set.

**Proof.** Suppose $\hat{P}(G) = (T, f')$ and $P(G) = (V, f)$.

Let $T' \subseteq T$ be any set. Then let $S$ be the set of vertices labeled with tags in $T'$. The sum over all vertices in $S$ of the vertex’s degree minus one is equal to $2|T'|$. This is because each tag contributes either one degree-3 vertex to $S$ or two degree-2 vertices. A degree-2 vertex contributes 1 to this sum (so two degree-2 vertices contribute 2) while a degree-3 vertex contributes 2 to the sum. Therefore each tag in $T'$ contributes 2 to the sum, and so the sum in question is equal to $2|T'|$.

By Lemma 4.14 $S$ is a nonseparating independent set if and only if $f(T')$ is equal to this sum (or in other words $f(T') = 2|T'|$). But the condition that $f(T') = 2|T'|$ is the definition of $T'$ being a matching of $(V, f)$.

Thus we have, as desired, that $T'$ is a matching of $\hat{P}(G) = (T, f')$ if and only if the set of vertices $S$ labeled with tags in $T'$ is a nonseparating independent set.

4.3 $\{(1, 2, 3), N\}$-TRVB is polynomial-time solvable

We prove that $\{(1, 2, 3), N\}$-TRVB is polynomial-time solvable by reducing to the Nonseparating Independent Tag Set problem.

**Theorem 4.15.** $\{(1, 2, 3), N\}$-TRVB is polynomial-time solvable.

**Proof.** The following is a procedure for solving $\{(1, 2, 3), N\}$-TRVB in polynomial time:
Take as input a multigraph $G_0$ whose breakable vertices each have degree at most 3. We can modify $G_0$ to get an equivalent graph $G_1$ (with respect to the TRVB problem) by inserting two unbreakable degree-2 vertices into every edge.

Next, convert graph $G_1$ into tagged graph $G_2$ as follows. Duplicate $G_1$ and include both copies into $G_2$. For each breakable degree-3 vertex in $G_1$, create two tags, assigning each copy of the vertex one of the tags. For each breakable degree-2 vertex in $G_1$, create one tag and assign it to both copies of the vertex.

Suppose $G_1$ had $n$ vertices and $m$ edges. Then use the polynomial-time algorithm for the Nonseparating Independent Tag Set problem to determine whether there exists a set of tags of size $m + 1 - n$ in $G_2$ such that the associated vertices make a nonseparating independent set. Output the answer of this subroutine as the answer of the $(\{1, 2, 3\}, \mathbb{N})$-TRVB problem.

Next, we will show that the above procedure is correct, or in other words that the TRVB problem of $G_1$ can be solved if and only if $G_2$ contains a set of $m + 1 - n$ tags whose corresponding vertices form a nonseparating independent set.

Suppose first that the TRVB problem of $G_1$ can be solved by breaking the vertices in some set $S$. We will show that in that case $G_2$ contains a set of $m + 1 - n$ tags whose corresponding vertices form a nonseparating independent set. Suppose the solution of the TRVB problem breaks $b_3$ vertices of degree 3, $b_2$ vertices of degree 2, and $b_1$ vertices of degree 1. Breaking a vertex does not affect the number of edges in a graph, but increases the number of vertices by the degree minus one. Thus the final vertex count in the graph is $b_2 + 2b_3 + n$. Since the final graph is a tree, the number of vertices is one more than the number of edges. Thus $b_2 + 2b_3 + n = m + 1$, or in other words $b_2 + 2b_3 = m + 1 - n$.

Furthermore, applying Lemma 4.3 tells us that $S$ is a nonseparating independent set in $G_1$. Then consider the copies of $G_1$ making up $G_2$ and let $S_2$ consist of the two copies of $S_1$ with all degree-1 vertices removed. Since $S_1$ is a nonseparating independent set in $G_1$ we know that $S_2$ is a nonseparating independent set in $G_2$. There are $b_2$ tags of degree-2 vertices in $S_2$ (with the tags shared across the two halves of $G_2$) and $2b_3$ tags of degree-3 vertices in $S_2$ (with one tag in each half of $G_2$ per degree-3 vertex in $S_1$). Thus the total number of tags for vertices in $S_2$ is $b_2 + 2b_3 = m + 1 - n$. The set $T$ of these tags is a solution to the Nonseparating Independent Tag Set instance consisting of $G_2$ and $m + 1 - n$.

Next suppose that we find a set of tags in $G_2$ of size $m + 1 - n$ whose corresponding vertex set is a nonseparating independent set. Suppose that there are $t_2$ tags for degree-2 vertex pairs, and $t_3$ tags for degree-3 vertices (with $t_2 + t_3 = m + 1 - n$). Then among the two halves of $G_2$ (each of which is a copy of $G_1$), one half will have a nonseparating independent set of $t_2$ degree-2 vertices and at least $\frac{t_3}{2}$ degree-3 vertices.

We can convert this into a nonseparating independent set $S$ in $G_1$ with $t_2$ degree-2 vertices and $\frac{t_3}{2} \geq \frac{3}{2} t_3$ degree-3 vertices. By Lemma 4.3, the graph formed by breaking the vertices of $S$ in $G_1$ is connected. This graph has the same number of edges as $G_1$ (in particular $m$), but has a number of vertices equal to $n + 2 \times \frac{t_3}{2} + t_2 = n + t_2 + t_3$ (since breaking a degree-3 vertex increases the number of vertices by two and breaking a degree-2 vertex increases the number of vertices by one). The difference between the number of vertices and edges is $(n + t_2 + t_3) - m = (n + m + 1 - n) - m = 1$. Thus we see that the resulting graph is connected and has at least one more vertex than it has edges. The only way this is possible is if the resulting graph is a tree. In other words, breaking the vertices in $S$ solves the TRVB problem on $G_1$.

We have shown that the TRVB instance that we wish to solve has a solution if and only if the Nonseparating Independent Tag Set instance that we actually solve has a solution. Therefore we have described a correct algorithm.
By Lemma 3.2, we can also conclude that

**Corollary 4.16.** $(B,U)$-TRVB is polynomial-time solvable if $B \subseteq \{1, 2, 3\}$. So are Planar $(B,U)$-TRVB, Graph $(B,U)$-TRVB, and Planar Graph $(B,U)$-TRVB.

5 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 5.1.** 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; though any such graph will still yield a “no” answer to the Tree-Residue Vertex-Breaking problem.

We begin with the proof idea in Section 5.1 and proceed through the details in Section 5.2

5.1 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.

Before we introduce the precise scoring mechanism, we can describe what aspects of the graph are being scored. Consider the state of the graph after breaking some number of vertices and in this graph consider one specific vertex. 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” separated by the edges leading to higher degree neighbors. For example, in Figure 11, 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.

Suppose we have a set $S$ of vertices such that breaking those vertices leaves the graph as a tree. We can choose to break these vertices in order starting on the exterior of the graph and moving inward. In particular, we can do this by repeatedly doing the following. Consider the external face of the graph. Either the graph is a tree (i.e. we have already broken all the vertices in $S$) or the external face is actually a cycle. Since every cycle includes a vertex in $S$ (otherwise that cycle
Figure 11: 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.

would remain after breaking every vertex in $S$), we can choose a vertex from $S$ on the external face of the graph and break that vertex next.

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

As it turns out, the final graph (a tree) cannot have disproportionately many small bundles, so we will use the intuition that the goal of breaking the vertices is to increase bundle sizes. To apply this intuition, we consider the effect of breaking some vertex of degree $d$ on the bundle sizes in the graph. It must be the case that any vertex in $S$ on the external face has exactly two edges which border this face. The remaining $d-2$ edges 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. Thus, breaking the vertex will necessarily create $d-2$ new bundles of size 1. Applying the intuition described above, we see that creating $d-2$ bundles of size one can be thought of as the “cost” of breaking the vertex. The “benefit” of breaking the vertex is that each of the two edges which were on the external face is now each added to a bundle, thereby increasing the size of that bundle. In summary, the breaking of a vertex can increase the sizes of two bundles by 1 each at the cost of creating $d-2 \geq 4$ new bundles of size 1.

Having said that, we can now specify how we score a bundle. If a bundle has a size of 1, we assign the bundle a score of $-1$. Otherwise, if the bundle has size $n$, we assign the bundle a score of $n-1$. If we do this, then the benefit of increasing a bundle’s size by 1 corresponds to a score increase of at most 2 (when increasing a bundle from size 1 to size 2). On the other hand the cost of adding a bundle of size 1 corresponds to a score decrease of 1. Thus, the effect of breaking one vertex is to increase the score by at most 2 twice and decrease the score by 1 at least 4 times. In other words, the score must either decrease or stay the same.

Then since the initial graph has no bundles (and therefore score 0), the final score of the tree that remains after breaking all vertices in $S$ cannot be positive. On the other hand, the final graph is a tree all of whose non-leaves have degree at least 5. It is possible to show that with the specific scoring mechanism described above, the score of such a tree is always positive. This is a contradiction, giving us our desired result.
5.2 Proof

In this section, we will follow the proof outline given in the previous section. We begin with a sequence of definitions leading to a formal definition of the scoring function used above.

Throughout this section, we will be considering a planar graph $G$ whose breakable vertices each have degree at least 6 and whose unbreakable vertices each have degree at least 5.

**Definition 5.2.** We say that $G'$ is a state of $G$ if we can obtain $G'$ by breaking some vertices of $G$.

We choose a particular planar embedding for $G$ to be the canonical planar embedding for $G$. When a vertex is broken in some state $G'$ of $G$, the new state $G''$ can inherit a planar embedding from $G'$ in the natural way: all vertices and edges unaffected by the vertex-breaking are embedded in the same place while the new vertices are placed so that the order of edges around each vertex is preserved. For example, breaking the center vertex in the planar embedding shown in the left part of Figure 12 would yield the planar embedding shown in the center of the figure rather than the right part. Then every state of $G$ can (via a sequence of states) inherit the canonical planar embedding for $G$. We will then use this embedding as the canonical embedding for the state of $G$. With that done, we no longer have to specify which planar embedding we are using for the states of $G$: we will always use the canonical planar embedding.

![Figure 12: Breaking the top vertex in the first planar embedding should yield the second planar embedding rather than the third in order to maintain the order of edges around the bottom vertex.](image)

**Definition 5.3.** We call $S$ a contiguous set of edges at $x$ if $S$ is a set of edges all sharing endpoint $x$ and we can proceed clockwise around $x$ starting and ending at some edge in $S$ such that the edges encountered are exactly those in $S$. Then a bundle at vertex $x$ is a non-empty maximal contiguous set of edges at $x$ whose other endpoints have degree 1.

Define the score of a bundle at vertex $x$ to be $-1$ if the bundle has size 1 and $n - 1$ if the bundle has size $n > 1$. Define the score of $x$ to be the cumulative score of all the bundles at $x$. Define the score of a state $G'$ of $G$ to be the cumulative score of all the vertices in $G'$.

Suppose for the sake of contradiction that $S$ is a set of breakable vertices in $G$ such that breaking the vertices in $S$ yields state $T$ which is a tree. In other words, suppose that there exists a solution to the instance $G$ of Planar Graph ($\{6, 7, 8, \ldots\}, \{5, 6, 7, \ldots\}$)-TRVB.

We begin by showing one side of the contradiction: that the score of $T$ is positive. To do this, we introduce the following lemma about trees:

**Lemma 5.4.** If $T'$ is a tree with at least 2 vertices, then

$$2(\text{number of leaves of } T') + (\text{number of degree-2 vertices in } T') > (\text{number of edges in } T').$$
Proof. Define $n_1(T')$ to be the number of leaves in $T'$, define $n_2(T')$ to be the number of degree-2 vertices in $T'$, and define $n_e(T')$ to be the number of edges in $T'$. Then we wish to show that for any tree $T'$ with at least 2 vertices, $2n_1(T') + n_2(T') > n_e(T')$. We will prove this by induction on the number of vertices in $T'$.

First consider the base case: if $X'$ is a tree on 2 vertices, then $X'$ contains exactly one edge (between its two vertices) so the number of leaves of $X'$ is 2, the number of degree-2 vertices is 0 and the number of edges is 1. We see then that $2n_1(X) + n_2(X) = 2 \times 2 + 0 = 4 > 1 = n_e(X)$ as desired.

Next suppose that for any tree $X$ on $i-1$ vertices it is the case that $2n_1(X) + n_2(X) > n_e(X)$. Let $X'$ be any tree on $i$ vertices and let $v$ be a leaf of $X'$. The graph $X' - \{v\}$ is a tree with $i-1$ vertices, so we can apply the inductive hypothesis: $2n_1(X' - \{v\}) + n_2(X' - \{v\}) > n_e(X' - \{v\})$.

Let $u$ be the sole neighbor of $v$ in $X$. The value $2n_1(X') + n_2(X')$ (twice the number of leaves plus the number of degree-2 vertices) changes as we go from $X' - \{v\}$ to $X'$ due to the change in degree of $u$ and the addition of $v$. Adding a neighbor to $u$ either converts $u$ from being a leaf to a degree-2 vertex, converts $u$ from being a degree-2 vertex to a degree-3 vertex, or converts $u$ from being a vertex of degree $n > 2$ to a vertex of degree $n+1$. In all cases, the value $2n_1(X') + n_2(X')$ decreases by at most one due to the change in degree of $u$. The addition of the new leaf $v$, on the other hand, increases this value by 2. Thus the overall increase of the value $2n_1(X') + n_2(X')$ when going from tree $X' - \{v\}$ to tree $X'$ is at least 1. In other words, we have that $2n_1(X') + n_2(X') \geq 2n_1(X' - \{v\}) + n_2(X' - \{v\}) + 1$.

Note also that the number of edges in $X'$ is one more than the number of edges in $X' - \{v\}$ (i.e. $n_e(X') = n_e(X' - \{v\}) + 1$).

Putting this all together, we see that

$$2n_1(X') + n_2(X') \geq 2n_1(X' - \{v\}) + n_2(X' - \{v\}) + 1 > n_e(X' - \{v\}) + 1 = n_e(X').$$

As desired, we have shown for any tree $X'$ with $i$ vertices that $2n_1(X') + n_2(X') > n_e(X')$ concluding the inductive step. By induction, we have shown that for any tree $T'$ with at least 2 vertices, $2n_1(T') + n_2(T') > n_e(T')$.

Lemma 5.5. The score of $T$ is positive.

Proof. Every vertex in $T$ is either a vertex originally in $G$ or a new vertex created by the breaking of some vertex in $S$. Vertices in $G$ have degree at least 5 and vertices created by the breaking of a vertex have degree 1. Thus every vertex in $T$ that is not a leaf is a vertex originally in $G$. Let $T'$ be the tree formed by removing every leaf from $T$. Notice that the vertices in $T'$ are exactly the vertices in $G \setminus S$.

Since every tree has at least one vertex of degree at most 1 and $G$ does not, we know that $G$ is not a tree. Thus $G \neq T$ and so $S \neq \emptyset$. Consider any vertex $v \in S$; $v$ has at least 6 neighbors, none of which can be in $S$ (since then breaking $S$ disconnects the graph). Thus the number of vertices in $T'$ is at least 6.

Below, we will show that the score of $T$ is

$$-2 \text{(number of edges in } T') + 4 \text{(number of leaves of } T') + 2 \text{(number of degree-2 vertices in } T').$$

But since $T'$ is a tree with at least 6 vertices, the previous lemma applies to show that

$$2 \text{(number of leaves of } T') + \text{(number of degree-2 vertices in } T') > \text{(number of edges in } T').$$

Simply rearranging (and doubling) this inequality immediately shows that the score of $T$ is positive. Thus, all that is left is to show that the expression given above for the score of $T$ is correct.
The score of $T$ is the sum over all vertices $x$ in $T$ of the score of $x$. If $x$ is not in $T'$, then $x$ is a leaf of $T$. If $x$ has a degree-1 neighbor in $T$, then the connected component of $x$ in $T$ would consist entirely of just those two vertices and as a result, $T$ would have no non-leaf nodes. This cannot be the case since $|T'| \geq 6$. Thus, $x$ has no degree-1 neighbors, and therefore there are no bundles at $x$. As a result, the score of $x$ is 0. Thus, the score of $T$ is the sum over all vertices $x$ in $T'$ of the score of $x$.

For $x \in T'$, define $d(x)$ to be the degree of $x$ in tree $T'$. For any vertex $x \in T'$, we can lower-bound the score of $x$ using casework:

- $d(x) = 0$. This would imply that $x$ is the only vertex in $T'$, directly contradicting the fact that $|T'| \geq 6$. Thus, we can conclude that this case is impossible.

- $d(x) = 1$. The vertex $x$ in has exactly one edge in $T$ leading to a non-leaf neighbor. This necessarily implies that all the other edges incident on $x$ form a bundle. Notice that $x$ has degree at least 5 in $T$ since it is a vertex that was originally in $G$. Thus, the score of this bundle is then at least 3, and so the score of $x$ is also at least $3 = 4 - d(x)$.

- $d(x) = 2$. The vertex $x$ has exactly two edges in $T$ leading to non-leaf neighbors. These two edges separate all of the other edges incident on $x$ into at most two bundles. Notice that $x$ has degree at least 5 in $T$ since it is a vertex that was originally in $G$. Thus there are at least 3 edges in the (at most) two bundles at $x$. If there is one bundle, then the bundle has size at least 3 and score at least 2. If there are two bundles, then the total size of the two bundles is at least 3, implying that the minimum possible total score of the two bundles is 0 (which occurs in the case that one bundle has size 1 and the other has size 2). In all cases, the score of $x$ is at least $0 = 2 - d(x)$.

- $d(x) > 2$. The vertex $x$ has exactly $d(x)$ edges leading to non-leaf neighbors. These $d(x)$ edges separate all of the other edges incident on $x$ into at most $d(x)$ bundles. Each bundle has a score of at least $-1$, so $x$ has a score of at least $-d(x)$.

If we use $1_X$ to represent the indicator function (which outputs 1 if $X$ is true and 0 otherwise), then the above results can be summarized as follows: the score of $x \in T'$ is bounded below by $(-d(x)) \times 1_{d(x) \geq 2} + (4 - d(x)) \times 1_{d(x) = 1} + (2 - d(x)) \times 1_{d(x) = 2}$. Equivalently, we have that the score of $x \in T'$ is bounded below by $-d(x) + 4 \times 1_{d(x) = 1} + 2 \times 1_{d(x) = 2}$.

Adding this up, we see that the score of $T$ is at least

$$\sum_{x \in T'} (-d(x) + 4 \times 1_{d(x) = 1} + 2 \times 1_{d(x) = 2}) = -\sum_{x \in T'} d(x) + 4 \sum_{x \in T'} 1_{d(x) = 1} + 2 \sum_{x \in T'} 1_{d(x) = 2}.$$ 

Since $\sum_{x \in T'} d(x)$ is the total degree of vertices in tree $T'$, this value is twice the total number of edges in $T'$. The terms $\sum_{x \in T'} 1_{d(x) = 1}$ and $\sum_{x \in T'} 1_{d(x) = 2}$ are the number of leaves and number of degree-2 vertices in $T'$.

Thus the score of $T$ is at least

$$-2(\text{number of edges in } T') + 4(\text{number of leaves of } T') + 2(\text{number of degree-2 vertices in } T').$$

As argued above, this implies our desired result: that the score of $T$ is positive.  

Next, we proceed to the other side of the contradiction: showing that the score of $T$ is non-positive.

To begin, we define an ordering $s_1, \ldots, s_{|S|}$ of the vertices in $S$ as follows:
Definition 5.6. Let $G_0 = G$. Then for $i = 1, \ldots, |S|$, define $s_i$ and $G_i$ as follows: Let $s_i$ be any vertex of $S$ that is on the boundary of the external face of $G_{i-1}$ and let $G_i$ be $G_{i-1}$ with vertex $s_i$ broken.

Lemma 5.7. Definition 5.6 is well defined.

Proof. Notice that once a vertex is broken, it is no longer in the graph. Thus, it is impossible for the procedure given in Definition 5.6 to assign some element of $S$ to be both $s_i$ and $s_j$ for $i \neq j$.

With that said, in order to conclude that Definition 5.6 is well defined, it is sufficient to show that at each step, a choice of $s_i$ satisfying the conditions given in the definition is possible.

Fix $i \in \{1, \ldots, |S|\}$. Notice that $G_{i-1}$ is not a tree since that would mean that breaking a proper subset $\{s_1, \ldots, s_{i-1}\}$ of $S$ in $G$ yields a tree (in which case breaking the rest of $S$ would disconnect $G$). From this, we can conclude that $G_{i-1}$ has both an external face and at least one internal face.

The points not in the external face form some set of connected regions; furthermore, this set is not empty since there is at least one internal face. The boundary of $R$ must consist of a cycle of edges separating the external face from the internal faces inside $R$. This cycle of edges must contain at least one vertex of $S$ since otherwise the cycle would remain in $T$ after breaking every vertex of $S$ in $G$. Furthermore, this cycle is a subset of the boundary of the external face.

Thus, when Definition 5.6 says to choose $s_i$ to be any vertex of $S$ that is on the boundary of the external face of $G_{i-1}$, this is well defined.

Definition 5.8. We say that edge $e$ is an external edge in $G_i$ if $e$ is an edge of $G_i$ with the external face on both sides. We say that $e$ is a boundary edge of $G_i$ if the external face is on exactly one side of $e$. Finally, we say that $e$ is an internal edge of $G_i$ if the external face is on neither side of $e$.

Lemma 5.9. Consider any edge $e$ incident on $s_i$ in graph $G_{i-1}$ and let $x$ be the other endpoint. When converting $G_{i-1}$ into $G_i$ by breaking $s_i$,

- if $e$ is a boundary edge, then $e$ either joins one previously existing bundle at $x$ or becomes a new bundle at $x$ of size 1.
- if $e$ is an internal edge, then $e$ becomes a new bundle at $x$ of size 1.

Proof. We begin with a proof by induction that every degree-1 vertex in $G_i$ is inside the external face for $i \in \{0, \ldots, |S|\}$. Certainly, it is the case that every degree-1 vertex in $G = G_0$ is inside the external face since $G$ has no degree-1 vertices. Then suppose all the degree-1 vertices of $G_{i-1}$ are inside the external face of $G_{i-1}$ for some $i$. We obtain $G_i$ from $G_{i-1}$ by breaking vertex $s_i$ on the boundary of the external face of $G_{i-1}$. As a result, the external face in $G_i$ is equal to the union of the faces touching $s_i$ in $G_{i-1}$ (including the external face). Thus, every degree-1 vertex that was inside the external face of $G_{i-1}$ is still inside the external face. Furthermore, every new degree-1 vertex (formed by breaking $s_i$) is also created inside the external face of $G_i$. By induction, we have our desired result: for $i \in \{0, \ldots, |S|\}$, every degree-1 vertex in $G_i$ is inside the external face.

Notice that $x$ must have at least one neighbor other than $s_i$ that has degree more than 1 in $G_{i-1}$ because otherwise breaking $s_i$ would disconnect $x$ and its neighbors from the rest of the graph. This means that if we start at edge $e$ and go clockwise around $x$, we will eventually encounter some edge leading to a neighbor with degree more than 1. Call this edge $e_+$. Similarly, starting at edge $e$ and going counterclockwise around $x$, we can let the first encountered edge leading to a neighbor with
degree more than 1 be \( e_- \). There may or may not be a bundle at \( x \) between \( e \) and \( e_+ \). Similarly there may or may not be a bundle at \( x \) between \( e \) and \( e_- \). In any case, when \( s_i \) is broken, these bundles, if they exist, are merged, and edge \( e \) is added to the one resulting bundle.

Suppose that there is a bundle between \( e \) and \( e_+ \). This means that there are degree-1 vertices in the face that is clockwise from \( e_0 \) around \( x \). Then since degree-1 vertices are always found inside the external face, we can conclude that the face on that side of \( e \) is the external face of \( G_{i-1} \). Similarly, we can use the same logic to show that if there is a bundle between \( e \) and \( e_- \), then the region on the other side of \( e \) is the external face of \( G_{i-1} \). These results are shown in Figure 13.

![Figure 13](image)

Figure 13: This figure shows vertices \( s_i \) and \( x \) with edge \( e \) between them and edges \( e_- \) and \( e_+ \) at \( x \) as defined in the proof. Regions \( R_+ \) and \( R_- \) are the regions on the two sides of \( e \). The bundle shown in region \( R_+ \) at \( x \) can be present only if \( R_+ \) is the external face. Similarly, the bundle shown in region \( R_- \) at \( x \) can be present only if \( R_- \) is the external face.

Next, consider the two cases in which edge \( e \) is a boundary or internal edge. If \( e \) is an internal edge, then it has the external face on zero sides, and so neither the bundle at \( x \) between \( e \) and \( e_+ \) nor the bundle at \( x \) between \( e \) and \( e_- \) can exist. Then when \( s_i \) is broken, edge \( e \) forms a new bundle of size 1 at \( x \). If \( e \) is a boundary edge, then it has the external face on exactly one side. In this case, at most one of the two bundles at \( x \) (between \( e \) and \( e_+ \) or \( e \) and \( e_- \)) can exist. Then when \( s_i \) is broken, edge \( e \) either forms a new bundle of size 1 (if neither bundle existed) or is added to a previously existing bundle at \( x \) (if one of the two bundles existed). This is exactly what we wished to show.

**Lemma 5.10.** When going from \( G_{i-1} \) to \( G_i \), the score cannot increase.

**Proof.** The only vertices in \( G_i \) that are not in \( G_{i-1} \) are the degree-1 vertices which replace \( s_i \) when it is broken. The only vertex in \( G_{i-1} \) that is not in \( G_i \) is \( s_i \). We claim that the neighbors of all of these vertices have degree not equal to 1, and therefore that each of these vertices has no bundles and thus a score of 0. Suppose for the sake of contradiction that some one of these vertices which are present in one graph but not the other has a neighbor of degree 1. This means that either a degree-1 vertex in \( G_i \) or \( s_i \) in \( G_{i-1} \) has a degree-1 neighbor. If \( s_i \) has a degree-1 neighbor, then when it is broken, one of the degree-1 vertices replacing it inherits that neighbor. Thus in all cases, some two degree-1 vertices in \( G_i \) are neighbors. These two vertices form a connected component,
implying that $G_i$ is not connected. This contradicts the fact that $S$ is a solution to the TRVB instance, so as desired, none of the vertices in question have degree-1 neighbors.

Thus, since every vertex in one of $G_{i-1}$ and $G_i$ but not both has a score of 0, the difference in score between $G_{i-1}$ and $G_i$ is equal to the cumulative difference in score over all vertices that are in both of these graphs. Suppose $x$ is a vertex of both graphs that does not neighbor $s_i$. Breaking $s_i$ does not affect $x$ or the degrees of the neighbors of $x$. Thus the bundles at $x$ remain the same in $G_{i-1}$ and $G_i$. In other words, there is no change in score at vertex $x$.

Then to compute the difference in scores between $G_{i-1}$ and $G_i$, we must simply compute the cumulative difference in scores over all vertices that neighbor $s_i$. If the edge between $s_i$ and $x$ is an internal edge, then by the previous lemma, the change in bundles at $x$ between $G_{i-1}$ and $G_i$ is that a new bundle of size 1 is added. This decreases the score of $x$ by 1. If the edge between $s_i$ and $x$ is a boundary edge, then by the previous lemma, the change in bundles at $x$ between $G_{i-1}$ and $G_i$ is either that a new bundle of size 1 is added or that the size of some one bundle is increased by 1. Thus, the score of $x$ either decreases by 1 (if a new bundle is added), increases by 2 (if a bundle of size 1 becomes a bundle of size 2), or increases by 1 (if a bundle of size at least 2 increases in size by 1). Below, we will show that exactly two of the edges incident on $s_i$ in $G_{i-1}$ are boundary edges and that the rest are internal edges. Since the degree of $s_i$ is at least 6, this means that exactly two neighbors of $x$ will have their score increase by at most 2 and that all the other neighbors (of which there are at least 4) will have their score decrease by 1. The change in score between $G_{i-1}$ and $G_i$ is then an increase of at most 4 plus a decrease of at least 4. In other words, when going from $G_{i-1}$ to $G_i$, the score cannot increase, which is exactly the statement of this lemma.

All that’s left is to show that exactly two of the edges incident on $s_i$ in $G_{i-1}$ are boundary edges and that the rest are internal edges.

Consider the faces which touch at $s_i$ in $G_{i-1}$. Since $s_i$ was chosen to be on the external face, one of these faces is the external face. Let this face be $F_0$, and let the other faces clockwise around $s_i$ be $F_1,F_2,...,F_{d-1}$ where $d$ is the degree of $s_i$ in $G_{i-1}$. Let $e_j$ be the edge separating face $F_j$ from the previous face clockwise around $s_i$. See Figure 14 for an example. Finally, let $x_j$ be the endpoint of $e_j$ other than $s_i$.

![Figure 14](image-url)

Figure 14: This figure shows an example vertex $s_i$ of $G_{i-1}$ together with the edges and faces meeting at $s_i$.

Since $G_i$ is connected, there exists a path from $x_0$ to $x_1$ in $G_i$. This path will also exist in $G_{i-1}$
since a path cannot use any vertices of degree 1 and every vertex in $G$ whose degree is not 1 is also in $G_{i-1}$. There is another path in $G_{i-1}$ from $x_0$ to $x_1$: namely the path $x_0, s_i, x_1$. Together, these two paths form a cycle in $G_{i-1}$ including the two edges $e_0$ and $e_1$. Either the faces that are between $e_0$ and $e_1$ clockwise around $s_i$ (in particular face $F_0$) or counterclockwise around $s_i$ (all the other $F_j$s) must be on the interior of this cycle. Since $F_0$ is the external face, we know that it cannot be on the interior of a cycle. Thus each other $F_j$ must be on the interior of the cycle and therefore cannot equal the external face. Notice that edges $e_0$ and $e_1$ have the external face on exactly one side and that each other edge $e_j$ has the external face on neither side. As desired, we have shown that exactly two of the edges incident on $s_i$ in $G_{i-1}$ are boundary edges and that the rest are internal edges.

Lemma 5.11. The score of $T$ is not positive.

Proof. $G_0 = G$ has no degree-1 vertices and therefore has no bundles. Thus, the score of $G_0$ is 0. By the previous lemma, the score of $G_i$ is non-increasing as a function of $i$. We can immediately conclude that the score of $G_{|S|}$, the graph formed by breaking vertices $s_1, \ldots, s_{|S|}$ is not positive. But $s_1, \ldots, s_{|S|}$ are all the vertices in $S$, so $G_{|S|} = T$ and as desired, the score of $T$ is not positive.

Notice that we have seen two directly contradictory lemmas: we have shown that the score of $T$ is both positive and not positive. By contradiction, we can conclude that $S$, the solution to Planar Graph ($\{6, 7, 8, \ldots\}, \{5, 6, 7, \ldots\}$)-TRVB instance $G$, cannot exist. Thus, we have that

Lemma 5.12. Planar Graph ($\{6, 7, 8, \ldots\}, \{5, 6, 7, \ldots\}$)-TRVB is polynomial-time solvable

Proof. We have shown that for any instance of Planar Graph ($\{6, 7, 8, \ldots\}, \{5, 6, 7, \ldots\}$)-TRVB, the correct answer is “no.” Thus rejecting all inputs (a polynomial-time algorithm) solves Planar Graph ($\{6, 7, 8, \ldots\}, \{5, 6, 7, \ldots\}$)-TRVB.

From this, we obtain our desired result.

Theorem 5.13. 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.

Proof. This follows immediately from the previous lemma together with Lemma 3.2.

6 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 6.1 and then reduce from this problem to Planar ($\{k\}, \{4\}$)-TRVB for any $k \geq 4$ in Section 6.2. 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.

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

The following problem was shown NP-complete in [3]:

Problem 5. 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 instances 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 6.1.** 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 6.** 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.

The goal of this section is to prove that this problem is NP-hard. To this purpose, consider the following definition and lemmas:

**Definition 6.2.** Define simplifying \( G \) over edge \((u,v)\) to be the following operation: remove all edges \((u',v)\) and \((u,v')\) from \( G \) and then contract edge \((u,v)\). The resulting graph has one new vertex instead of \( u \) and \( v \); this vertex inherits the inward oriented edges of \( u \) and inherits the outward oriented edges of \( v \). The inward oriented edges of \( v \) and outward oriented edges of \( u \) are removed from the graph.

**Lemma 6.3.** If \((u,v)\) is an edge of directed graph \( G \) and either \( u \) has outdegree 1 or \( v \) has indegree 1, then simplifying \( G \) over \((u,v)\) maintains the Hamiltonicity of \( G \).

**Proof.** Let \( G' \) be the graph that results from simplifying \( G \) over edge \((u,v)\) and let \( w \) be the vertex in \( G' \) that replaces \( u \) and \( v \). Any Hamiltonian cycle \( x_1, x_2, \ldots, x_{n-2}, u, v \) in \( G \) using edge \((u,v)\) corresponds with Hamiltonian cycle \( x_1, x_2, \ldots, x_{n-2}, w \) in \( G' \). And any Hamiltonian cycle \( x_1, x_2, \ldots, x_{n-2}, w \) in \( G' \) corresponds with Hamiltonian cycle \( x_1, x_2, \ldots, x_{n-2}, u, v \) in \( G \) using edge \((u,v)\). Thus there is a bijection between Hamiltonian cycles of \( G' \) and Hamiltonian cycles of \( G \) using edge \((u,v)\).

But if either \( u \) has outdegree 1 or \( v \) has indegree 1, then every Hamiltonian cycle in \( G \) must use edge \((u,v)\), and so the Hamiltonian cycles of \( G \) using edge \((u,v)\) are all the Hamiltonian cycles of \( G \). Thus there is a bijection between Hamiltonian cycles of \( G' \) and Hamiltonian cycles of \( G \), and so the numbers of Hamiltonian cycles in \( G \) and \( G' \) are the same. As desired, \( G' \) is Hamiltonian if and only if \( G \) is.

**Lemma 6.4.** If \((u,v)\) is an edge of planar non-alternating directed graph \( G \), then simplifying \( G \) over \((u,v)\) maintains the planar non-alternating property of \( G \).

**Proof.** Let \( G' \) be the graph that results from simplifying \( G \) over edge \((u,v)\). Starting with a planar non-alternating embedding of \( G \), the corresponding planar embedding of \( G' \) will also be non-alternating. We prove this below.
If \( x \) is a vertex of \( G \) that is not \( u \) or \( v \), then in the planar non-alternating embedding \( x \) will have all the inward oriented edges in one contiguous section. The simplification of \( G \) over \((u, v)\) will at most affect \( x \) by removing some edges incident on \( x \). In no case does this introduce alternation of inward and outward oriented sections to \( x \). Thus \( x \) is non-alternating in the planar embedding of \( G' \).

If \( x \) is the new vertex introduced due to the simplification of \( G \) over \((u, v)\), then \( x \) is non-alternating in the planar embedding of \( G' \) because (1) the inward oriented edges are all inherited from \( u \), (2) the outward oriented edges are all inherited from \( v \), and (3) the edges inherited from the two vertices by \( x \) can be separated into two contiguous sections.

As desired, this shows that \( G' \) is planar non-alternating.

We apply these lemmas to prove that the Planar Non-Alternating Indegree-2 Outdegree-2 Hamiltonicity Problem is NP-hard:

**Theorem 6.5.** The Planar Non-Alternating Indegree-2 Outdegree-2 Hamiltonicity Problem is NP-hard.

**Proof.** We prove this via the following reduction from the Planar Max-Degree-3 Hamiltonicity Problem. On input a planar graph \( G \) with all in- and out-degrees 1 or 2, repeatedly identify edges \((u, v)\) such that either \( u \) has outdegree 1 or \( v \) has indegree 1 and simplify \( G \) over \((u, v)\). Only stop once no such edges \((u, v)\) can be found, at which point output the resulting graph \( G' \).

First note that this algorithm runs in polynomial time since (1) simplification is a polynomial-time operation and (2) the number of simplifications of \( G \) is bounded above by the number of vertices in \( G \) since each simplification decreases the number of vertices by 1.

Suppose the input instance \( G \) is a “no” instance of the Planar Max-Degree-3 Hamiltonicity Problem. This means that \( G \) is not Hamiltonian. By repeated application of Lemma 6.3, \( G' \) is Hamiltonian if and only if \( G \) is Hamiltonian. Thus \( G' \) is not Hamiltonian and so \( G' \) is a “no” instance of the Planar Non-Alternating Indegree-2 Outdegree-2 Hamiltonicity Problem.

On the other hand, suppose the input instance \( G \) is a “yes” instance of the Planar Max-Degree-3 Hamiltonicity Problem. By repeated application of Lemma 6.3, \( G' \) is Hamiltonian if and only if \( G \) is Hamiltonian, so \( G' \) must have a Hamiltonian cycle. Below we show that all in- and out-degrees in \( G' \) are 2 and that \( G' \) is a planar non-alternating graph. Together, this is enough to imply that \( G' \) is a “yes” instance of the Planar Non-Alternating Indegree-2 Outdegree-2 Hamiltonicity Problem.

Since \( G' \) has a Hamiltonian cycle, no vertex in \( G' \) can have in- or out-degree 0. Furthermore, no vertex in \( G' \) can have in- or out-degree 1 because the reduction does not stop simplifying the graph until there are no in- or out-degree 1 vertices left. Thus every in- or out-degree in \( G' \) is at least 2. When simplifying a graph over an edge, every in- or out-degree in the resulting graph is less than or equal to some in- or out-degree in the initial graph. By repeatedly applying this rule, we see that every in- and out-degree in \( G' \) is at most the largest in- or out-degree in \( G \). But as \( G \) is a Planar Max-Degree-3 Hamiltonicity instance, the largest in- or out-degree in \( G \) is at most 2. Thus, we can conclude that every in- and out-degree in \( G' \) must be exactly 2.

By repeated application of Lemma 6.4, we know that provided the original graph \( G \) is a planar non-alternating graph, the final graph \( G' \) will be as well. But if \( G \) is a planar max-degree-3 graph, then every vertex in \( G \) is non-alternating in any planar embedding (since alternating vertices always have total degree at least 4). Thus, any planar embedding of \( G \) is a planar non-alternating embedding. We can therefore conclude that both \( G \) and \( G' \) are planar non-alternating graphs.

As desired, \( G \) is a “yes” instance of the Planar Max-Degree-3 Hamiltonicity Problem if and only if \( G' \) is a “yes” instance of the Planar Non-Alternating Indegree-2 Outdegree-2 Hamiltonicity
6.2 Reduction to Planar (\{k\}, \{4\})-TRVB for any \(k \geq 4\)

Consider the following definition:

**Definition 6.6.** For \(k \geq 4\), algorithm \(R_k\) defined below takes as input a planar non-alternating graph whose vertex in- and out-degrees all equal 2, and outputs an instance of Planar (\{k\}, \{4\})-TRVB.

Suppose that \(G\) is the graph being given as input to \(R_k\). Then to begin, we construct a labeled undirected multigraph \(M\) as follows.

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 \(m((u_2, v))\).

Finally, pick any specific vertex \(\hat{v}\) in \(G\). 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 \(R_k\).

In order to analyze the behavior of algorithm \(R_k\), it will be helpful to have the following definition:

**Definition 6.7.** We say that two edges in a planar non-alternating indegree-2 outdegree-2 graph are conflicting if they start or end at the same vertex. A Hamiltonian cycle in such a graph must contain exactly one out of every pair of conflicting edges.

**Lemma 6.8.** The output \(M'\) of \(R_k\) is a planar labeled multigraph whose vertices are all breakable with degree \(k\) or unbreakable with degree \(4\).

**Proof.** Define all variables used in the description of \(R_k\) as defined there. Because \(G\) is planar non-alternating, we can immediately conclude that multigraph \(M\) is planar as well (see Figure 15 for an example).

Consider any vertex \(m(v)\) in \(M\) (where \(v\) is a vertex of \(G\)). This vertex has exactly four neighbors: the vertices \(m(e)\) for every edge \(e\) in \(G\) that is incident on \(v\). Furthermore, this vertex is unbreakable.

Consider any vertex \(m((u, v))\) in \(M\) (where \((u, v)\) is an edge of \(G\)). This vertex has one edge to \(m(u)\), one edge to \(m(v)\), one edge to \(m((u, v'))\), and \(k - 3\) edges to \(m((u', v))\) (where \((u, v')\) and \((u', v)\) are the two edges in \(G\) conflicting with \((u, v))\). Thus the degree of this vertex is \(k\).

This shows that \(M\) consists of only degree-4 unbreakable vertices and degree-\(k\) breakable vertices. Thus, we have shown that \(M\) has exactly those properties that we are trying to show for \(M'\): \(M\) is a planar labeled multigraph whose vertices are all breakable with degree \(k\) or unbreakable with
degree 4. All that’s left is to show that the operation converting \( M \) to \( M' \) leaves these properties unchanged.

To convert \( M \) to \( M' \), vertex \( m(\hat{v}) \) is removed, and two edges \((m((u_1, \hat{v})), m((u_2, \hat{v})))\), and \((m((\hat{v}, w_1)), m((\hat{v}, w_2)))\) are added.

Note first that the four endpoints of these two edges are exactly the four neighbors of \( m(\hat{v}) \) in \( M \). Thus, each vertex in \( M \) other than \( m(\hat{v}) \) has the same degree in \( M' \): either the vertex was unaffected by the change from \( M \) to \( M' \) or a single edge was removed from the vertex and a single edge was added. Therefore the vertices of \( M' \) are all breakable with degree \( k \) or unbreakable with degree 4.

Next note that the two edges added to the multigraph are both already present. Increasing the multiplicity of an edge in a multigraph does not affect the planarity of the multigraph, and neither does removal of vertices and edges. Thus, the operation transforming \( M \) into \( M' \) maintains the planarity of the multigraph.

We can conclude that we have our desired properties: \( M' \) is a planar labeled multigraph whose vertices are all breakable with degree \( k \) or unbreakable with degree 4. This can be seen for the Figure 15 example in Figure 16.

The following is an additional, trivial, property of \( R_k \):

**Lemma 6.9.** \( R_k \) runs in polynomial time.

Consider the following lemma:

**Lemma 6.10.** Suppose \( R_k \) outputs \( M' \) on input \( G \) and there exists a solution to the TRVB problem on \( M' \). Then the set of edges \( e \) in \( G \) such that \( m(e) \) is not broken is a disjoint cycle cover of \( G \).

**Proof.** Consider any pair of conflicting edges \( e_1 \) and \( e_2 \) in \( G \) that share endpoint \( v \). There exists at least one edge in \( M \) between \( m(e_1) \) and \( m(e_2) \), and this edge is still present in \( M' \). Thus, in
Figure 16: If the $M$ produced by the algorithm is shown in Figure 15 (for $k = 4$), then shown here is one possible $M'$ (where the shown multigraph is $M'$ if $\hat{v}$ is chosen to be the bottom left vertex). Notice that $M'$ has our desire properties: $M'$ is a planar labeled multigraph whose vertices are all breakable with degree $k$ or unbreakable with degree 4.

order to avoid disconnecting that edge from the rest of the graph, either $m(e_1)$ or $m(e_2)$ must not be broken. $M$ also contains a cycle on three vertices $m(e_1)$, $m(e_2)$, and $m(v)$. If $v = \hat{v}$, then the third vertex is missing in $M'$, but in that case there is instead a cycle in $M'$ with just $m(e_1)$ and $m(e_2)$. In any case, $M'$ contains at least one cycle whose only breakable vertices are $m(e_1)$ and $m(e_2)$. In order for the resulting graph to be acyclic, at least one of these two vertices must be broken. This shows that in any solution to the TRVB problem on $M''$, exactly one out of every pair of conflicting edges $(e_1, e_2)$ has $m(e_i)$ broken.

Consider the set $C$ of edges $e$ in $G$ such that $m(e)$ is not broken. For every vertex $v$ of $G$, the two edges out of $v$ conflict and the two edges into $v$ conflict. Since every pair of conflicting edges $(e_1, e_2)$ has exactly one $m(e_i)$ broken, we conclude that $C$ contains one edge that enters $v$ and one that exists it. Thus $C$ is a disjoint cycle cover of $G$, as desired.

Based on this lemma, we can define the following correspondence:

**Definition 6.11.** For any solution of TRVB instance $M'$, define $C$ to be the disjoint cycle cover of $G$ consisting of edges $e$ such that $m(e)$ is not broken in the given solution of $M'$.

As per this definition, we can derive a disjoint cycle cover of $G$ from any solution to TRVB instance $M'$. Similarly, for any disjoint cycle cover of $G$, we can derive a candidate solution (though not necessarily an actual solution) for $M'$: simply break every vertex $m(e)$ where $e$ is an edge of $G$ that is not in the given disjoint cycle cover. Then for some suitable definition of “candidate solution,” there is a bijection between disjoint cycle covers of $G$ and candidate solutions of TRVB instance $M'$. We will show below that in fact, a disjoint cycle cover of $G$ is actually a Hamiltonian cycle if and only if the corresponding candidate solution for $M'$ is actually a solution. For example, see Figure 17.
Lemma 6.12. Suppose $R_k$ outputs $M'$ on input $G$. If there exists a solution to the TRVB problem on $M'$, then the corresponding cycle cover of $G$ is actually a Hamiltonian cycle.

Proof. Let $C$ be the disjoint cycle cover of $G$ consisting of edges $e$ such that $m(e)$ is not broken in the given solution of $M'$. We know that $C$ is a union of disjoint cycles and we wish to show that there is exactly one cycle in $C$. Let $v$ be a vertex in $G$ and let $C_v$ be the cycle in $C$ containing $v$. We will prove that $C$ contains exactly one cycle by proving that $C_v$ contains every vertex of $G$.

Let $M_{\text{solved}}'$ be the solved version of $M'$ (after breaking vertices) and let $M_{\text{solved}}$ be a version of $M$ in which the same vertices are broken. Consider the difference between $M$ and $M'$ from a connectivity standpoint. In $M$, vertex $m(\hat{v})$ connects its four neighbors, while in $M'$, these neighbors are instead connected in pairs with two edges. Thus, $M$ is at least as connected as $M'$. This connectivity pattern carries through to the solved versions of these multigraphs: $M_{\text{solved}}$ is at least as connected as $M'_{\text{solved}}$. Since $M'_{\text{solved}}$ is a tree, it is fully connected, and so $M_{\text{solved}}$ is also fully connected.

From this, we see that there exists a path in $M_{\text{solved}}$ from $m(v)$ to $m(v')$ for any vertex $v'$ in $G$. This path starts in $m(v)$, ends in $m(v')$, and passes through vertices that all have degree at least 2. Therefore every vertex in this path is a vertex from the original multigraph $M$ that was not broken. We prove below that every path in $M_{\text{solved}}$ using only vertices originally in $M$ which starts in $m(v)$ must end at a vertex of the form $m(x)$ where $x$ is a vertex or edge in cycle $C_v$. Since there exists a path in $M_{\text{solved}}$ using only vertices originally in $M$ from $m(v)$ to $m(v')$, we conclude that $v'$ is a vertex in $C_v$. Applying this to every vertex in $G$, we see that $C_v$ is a cycle containing every vertex in $G$, and therefore $C = C_v$ is a Hamiltonian cycle.

Consider any path in $M_{\text{solved}}$ using only vertices originally in $M$ which starts in $m(v)$. We prove by induction on the path length that this path ends at a vertex of the form $m(x)$ where $x$ is a vertex or edge in cycle $C_v$.

If the path length is zero, then the end vertex is $m(v)$, which is certainly of the correct form.

Next, suppose that the statement holds for all paths of length $i - 1$ or less. Then given a path of length $i$, we can apply the inductive hypothesis to this path without the last step. Thus we have
Lemma 6.13. Suppose \( R_k \) outputs \( M' \) on input \( G \) and there exists a Hamiltonian cycle in \( G \). Then the corresponding candidate solution of the TRVB instance \( M' \) is a solution.

Proof. Suppose that the Hamiltonian cycle of \( G \) is \( C \). Then let \( S \) be the set of vertices \( m(e) \) such that \( e \) is an edge of \( G \) not in \( C \). If \( m(e_1), m(e_2) \in S \), then \( e_1 \) and \( e_2 \) cannot be conflicting edges (since out of every pair of conflicting edges, one is in \( C \)). Therefore there is no edge between \( m(e_1) \) and \( m(e_2) \). Thus \( S \) contains no pair of adjacent vertices and is therefore an independent set in \( M' \).

Let \( A \) be any subset of \( S \). Define \( M'_A \) to be the multigraph derived from \( M' \) by removing the vertices in \( A \). Every vertex in \( M' \) is either equal to \( m(v) \) or adjacent to \( m(v) \) for some vertex \( v \) in \( G \). Then since \( A \subseteq S \) contains only vertices of the form \( m(e) \) where \( e \) is an edge of \( G \), we conclude that every vertex in \( M'_A \) is either equal to \( m(v) \) or adjacent to \( m(v) \) in \( G \). Then provided we show that every \( m(v) \) is in the same connected component of \( M'_A \), we will be able to conclude that \( M'_A \) is a connected graph. To show this, we prove that there exists a path in \( M'_A \) from any \( m(v_1) \) to any \( m(v_2) \) (with \( v_1 \) and \( v_2 \) vertices in \( G \)).

Consider the path \( v_1 = x_1, x_2, \ldots, x_l = v_2 \) from \( v_1 \) to \( v_2 \) in \( G \) which is part of Hamiltonian cycle \( C \) and excludes vertex \( \check{v} \). Then consider the following list of vertices:

\[
m(x_1), m((x_1, x_2)), m(x_2), m((x_2, x_3)), m(x_3), \ldots, m((x_{l-1}, x_l)), m(x_l).
\]

This list of vertices is a path in \( M \), so since \( m(\check{v}) \) is not in the list, it is also a path in \( M' \). Thus we have a path in \( M' \) from \( m(v_1) \) to \( m(v_2) \). No vertex in this path of the form \( m(x) \) where \( x \) is a vertex of \( G \) is in \( A \) simply because \( A \subseteq S \subseteq \{m(e) \mid e \text{ is an edge of } G\} \). Every vertex of the form \( m(e) \) where \( e \) is an edge of \( G \) has \( e \in C \), and so vertex \( m(e) \) is not in \( S \) (and consequentially not in \( A \)). Thus, no vertex in the path is in \( A \), and so every vertex in the path is present in \( M'_A \). Therefore this path also is a path from \( m(v_1) \) to \( m(v_2) \) in \( M'_A \).

As described above, this allows us to conclude that \( M'_A \) is connected. We see that for any subset \( A \subseteq S \), removing \( A \) from \( M' \) leaves the multigraph connected. Thus every \( A \subseteq S \) is not a separating set of \( M' \), and so \( S \) is a nonseparating set of \( M' \).

We have shown that \( S \) is a nonseparating independent set in \( M' \), so by Lemma 4.3, the multigraph that results from breaking the vertices of \( S \) in \( M' \) is connected.

Suppose \( G \) has \( n \) vertices. Then the number of edges in \( G \) is \( 2n \). The number of edges of \( G \) not in a Hamiltonian cycle is \( n \), so \(|S| = n\). Then the number of vertices in \( M \) is \( n + 2n = 3n \),
the total number of vertices and edges in $G$. The number of edges in $M$ is $\frac{n\times 4 + 2n\times k}{2} = n(k + 2)$. Transitioning from $M$ to $M'$ requires converting one vertex and four edges into zero vertices and two edges. Thus $M'$ has $3n - 1$ vertices and $n(k + 2) - 2$ edges.

Each vertex in $S$ has degree $k$, so breaking the $n$ vertices of $S$ in $M'$ increases the number of vertices in the resulting multigraph by $n(k-1)$. Then the multigraph that results from breaking the vertices of $S$ in $M'$ has $3n-1+n(k-1) = n(k+2)-1$ vertices and $n(k+2)-2$ edges. Furthermore, we showed above that this multigraph is connected. Since this multigraph is connected and has one more vertex than it has edges, we can conclude that it is a tree.

We have shown that if $G$ has a Hamiltonian cycle, breaking the vertices in $M'$ of set $S$ as defined above yields a tree. Thus we have that in the case that $G$ has a Hamiltonian cycle, the corresponding candidate solution of the TRVB instance $M'$ is a solution.

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

\begin{proof}
Consider the following reduction from Planar Non-Alternating Indegree-2 Outdegree-2 Hamiltonicity Problem to Planar $((\{k\}, \{4\})$-TRVB. On input a graph $G$, we first check whether $G$ is a planar non-alternating graph all of whose in- and out-degrees are 2. If yes, we run $R_k$ on input $G$ and output the result. Otherwise, we simply output any “no” instance of Planar $((\{k\}, \{4\})$-TRVB.

Since $R_k$ runs in polynomial time, the above is clearly a polynomial-time reduction. Furthermore, $R_k$ always outputs a planar labeled multigraph whose vertices are all breakable with degree $k$ or unbreakable with degree 4. As a result, in order to show that the above reduction is answer-preserving, it is sufficient to show that for all planar non-alternating graphs $G$ whose in- and out-degrees are 2, $G$ has a Hamiltonian cycle if and only if the corresponding output $M'$ of $R_k$ on input $G$, when interpreted as a TRVB instance, has a solution. This is exactly what we showed in the previous two lemmas.

Since the Planar Non-Alternating Indegree-2 Outdegree-2 Hamiltonicity Problem is NP-hard, we conclude that for any $k \geq 4$, Planar $((\{k\}, \{4\})$-TRVB is NP-hard.
\end{proof}

\section{Planar TRVB and TRVB are NP-complete with high-degree breakable vertices}

The goal of this section is to show that Planar $(B,U)$-TRVB and $(B,U)$-TRVB are NP-complete if $B$ contains any $k \geq 4$. To do this, we will show that Planar $((\{k\}, \emptyset)$-TRVB is NP-hard for any $k \geq 4$.

\begin{lemma}
For any $k \geq 4$, there exists a reduction from either Planar $((\{k\}, \{3\})$-TRVB or Planar $((\{k\}, \{4\})$-TRVB to Planar $((\{k\}, \emptyset)$-TRVB.
\end{lemma}

\begin{proof}
Below we will show that if $k \geq 4$, it is possible to simulate either a degree-4 unbreakable vertex or a degree-3 unbreakable vertex with a small gadget consisting of degree-$k$ breakable vertices. As a result, for every $k \geq 4$, we can construct a reduction from either Planar $((\{k\}, \{4\})$-TRVB or Planar $((\{k\}, \{3\})$-TRVB to Planar $((\{k\}, \emptyset)$-TRVB.

In particular, our reduction simply replaces every occurrence of an unbreakable degree-3 or degree-4 vertex with the corresponding gadget made of breakable degree-$k$ vertices. Provided we can design gadgets of constant size (with respect to the size of $G$, not with respect to $k$) whose behavior is the same as the vertex they are simulating, this reduction will be correct and will run in polynomial time.

To design the gadgets, we have two cases.
For $k = 4$, we use the gadget shown in Figure 18. Suppose that the gadget shown was included in a Tree-Residue Vertex-Breaking instance. In order to break the cycle between $P_0$ and $P_1$ without disconnecting the edges between them from the rest of the graph, exactly one of those two vertices $P_i$ must be broken. But then the neighbor $Q_i$ of $P_i$ cannot be broken without disconnecting the edge $(P_i, Q_i)$ from the rest of the graph. On the other hand, the node $Q_{1-i}$ cannot be broken either since breaking it would disconnect $P_{1-i}$ from the rest of the graph. Thus any valid solution of the Tree-Residue Vertex-Breaking instance must break neither $Q_i$ and exactly one $P_i$. The resulting graph connects the other four neighbors of the $Q_i$s without forming any cycles. In other words the behavior of this gadget in a graph is the same as the behavior of an unbreakable degree-4 vertex.

![Figure 18](image)

For $k \geq 5$, we use the gadget shown in Figure 19. In this gadget, breakable vertex $Q$ has $2a$ edges to other vertices $P_0, P_1, \ldots, P_{2a-1}$ in the gadget and $k - 2a$ edges out of the gadget. In addition, there are $k - 1$ edges between $P_{2i}$ and $P_{2i+1}$ for every $i$ in $\{0, 1, \ldots, a - 1\}$. Note that the degree of each vertex is $k$, as desired. When solving a graph containing this gadget, the cycle between $P_{2i}$ and $P_{2i+1}$ guarantees that exactly one of the two vertices must be broken. In order to not disconnect the other vertex from the rest of the graph, $Q$ cannot be broken in any valid solution. Thus, provided $a > 0$, every valid solution will break exactly one $P_{2i+j}$ with $j \in \{0, 1\}$ for each $i$ and will not break $Q$. If this is done, the part of the resulting graph corresponding to this gadget will connect the $k - 2a$ external neighbors of $Q$ to each other without forming any cycles. In other words the behavior of this gadget in a graph is the same as the behavior of an unbreakable degree-$(k - 2a)$ vertex.

![Figure 19](image)

Since $k \geq 5$, it is possible to choose $a > 0$ such that $k - 2a \in \{3, 4\}$. Then for every $k$, we are able to make a gadget to simulate either an unbreakable degree-4 vertex or an unbreakable degree-3 vertex. In all cases we can make the required gadgets, and so the reductions go through.

We already know that Planar ($\{k\}, \{4\}$)-TRVB is NP-hard from Section 6, so to obtain NP-hardness from the previous lemma, all that is left is to show that Planar ($\{k\}, \{3\}$)-TRVB is
NP-hard.

**Lemma 7.2.** Planar \((\{k\}, \{3\})\)-TRVB is NP-hard for any \(k \geq 4\).

**Proof.** We reduce from Planar \((\{k\}, \{4\})\)-TRVB to Planar \((\{k\}, \{3\})\)-TRVB.

Consider any unbreakable degree-4 vertex \(v\). We can replace \(v\) with two new degree-3 unbreakable vertices \(u\) and \(u'\) with edges between the two new vertices and the neighbors of \(v\) and an edge between \(u\) and \(u'\). Note that if we allocate two neighbors of \(v\) to each of \(u\) and \(u'\), we succeed in making \(u\) and \(u'\) have degree 3. Also note that it is possible to do this while maintaining the planarity of a multigraph. See Figure 20.

![Figure 20](image-url)  
Figure 20: 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.

Note that this pair of vertices “behaves” exactly the same as the original vertex did; in other words this change does not affect the answer to the Tree-Residue Vertex-Breaking question. As a result, applying this change to every unbreakable degree-4 vertex \(v\) converts a Planar \((\{k\}, \{4\})\)-TRVB instance into a Planar \((\{k\}, \{3\})\)-TRVB instance.

By Theorem 6.14 Planar \((\{k\}, \{4\})\)-TRVB is NP-hard for any \(k \geq 4\), and so the existence of this reduction proves that Planar \((\{k\}, \{3\})\)-TRVB is NP-hard for any \(k \geq 4\).

**Corollary 7.3.** Planar \((\{k\}, \emptyset)\)-TRVB is NP-hard for any \(k \geq 4\).

**Proof.** Lemmas 7.1 and 7.2, together with Theorem 6.14, allow us to conclude the desired result.

**Theorem 7.4.** 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 \(k \geq 4\).

**Proof.** By Lemma 3.2, there is a reduction from Planar \((\{k\}, \emptyset)\)-TRVB to Planar \((B,U)\)-TRVB if \(B\) contains \(k\). Thus, if \(k \geq 4\) and \(B\) contains \(k\), then Planar \((B,U)\)-TRVB is NP-hard. Lemma 3.2 also gives a reduction from Planar \((B,U)\)-TRVB to \((B,U)\)-TRVB, so we see that if \(k \geq 4\) and \(B\) contains \(k\), then \((B,U)\)-TRVB is also NP-hard.

Using Corollary 3.4 we see that as desired, if \(k \geq 4\) and \(B\) contains \(k\), then Planar \((B,U)\)-TRVB and \((B,U)\)-TRVB are both NP-complete.

**8 Graph TRVB is NP-complete with high-degree breakable vertices**

The goal of this section is to show that Graph \((B,U)\)-TRVB is NP-complete if \(B\) contains any \(k \geq 4\). To do this, we will show that Graph \((\{k\}, \emptyset)\)-TRVB is NP-hard for any \(k \geq 4\).

**Lemma 8.1.** Graph \((\{k\}, \{2\})\)-TRVB is NP-hard if \(k \geq 4\).
Proof. In Corollary 7.3 we saw that Planar ($\{k\}, \emptyset$)-TRVB is NP-hard if $k \geq 4$. We will reduce from this problem to Graph ($\{k\}, \{2\}$)-TRVB.

In order to do so, we must convert a given multigraph into a graph. One way to do this is to insert two degree-2 unbreakable vertices into every edge. After doing this, there will no longer be any duplicated edges or self loops, and so the result will be a graph. Furthermore, adding an unbreakable degree-2 vertex into the middle of an edge does not influence the answer to a Tree-Residue Vertex-Breaking question. Thus applying this transformation is a valid reduction.

We conclude that as desired, Graph ($\{k\}, \{2\}$)-TRVB is NP-hard if $k \geq 4$.

**Theorem 8.2.** Graph ($\{k\}, \emptyset$)-TRVB is NP-hard if $k \geq 4$.

**Proof.** In Lemma 8.1 we saw that Graph ($\{k\}, \{2\}$)-TRVB is NP-hard if $k \geq 4$. We wish to reduce from that problem to Graph ($\{k\}, \emptyset$)-TRVB.

In order to do this, we construct a constant sized (in the size of $G$) gadget using degree-$k$ breakable vertices that behaves the same as a degree-$2$ unbreakable vertex. Simply replacing every degree-$2$ unbreakable vertex with a copy of this gadget is a valid reduction, allowing us to conclude that Graph ($\{k\}, \emptyset$)-TRVB is NP-hard if $k \geq 4$.

The gadget is shown in Figure 21. The gadget consists of $2k - 2$ breakable vertices, each of degree $k$. Call them $P_1, P_2, \ldots, P_{k-2}$ and $Q_1, Q_2, \ldots, Q_k$. The gadget contains an edge between each pair $(P_i, Q_j)$ and an edge between each pair $(Q_i, Q_{i+1})$. Finally, both $Q_1$ and $Q_k$ will have one edge leaving the gadget.

![Figure 21](image.png)

Figure 21: A gadget simulating an unbreakable degree-2 vertex using only breakable degree-$k$ vertices arranged without self loops or duplicated edges.

In a solution to this gadget, either $P_1$ is broken or not. If $P_1$ is broken, then to avoid disconnecting the edge $(Q_i, P_1)$ from the rest of the graph, $Q_i$ must not be broken. But $(Q_1, Q_2, P_1)$ is a cycle for every $i$, so in order to avoid having that cycle in the final graph, $P_i$ must also be broken.

If $P_1$ is not broken, then either $Q_1$ or $Q_2$ must be broken (due to cycle $(Q_1, Q_2, P_1)$). Then if $Q_i$ is broken, $P_j$ must not be broken in order to avoid disconnecting edge $(Q_i, P_j)$ from the rest of the graph. This means that every $P_{j}$ will not be broken. In that case, however, the existence of cycle $(Q_{i_1}, P_1, Q_{i_2}, P_2)$ guarantees that either $Q_{i_1}$ or $Q_{i_2}$ will be broken for every pair $i_1, i_2$. In other words, at most one $Q_i$ can be unbroken. This means, however, that either both $Q_1$ and $Q_2$ or both $Q_3$ and $Q_4$ will be broken, in either case isolating an edge from the rest of the graph. Thus we see that this case is not possible.

We can conclude that the only solution to this gadget is to break all of the $P_i$s but none of the $Q_i$s, thereby connecting the external neighbors of $Q_1$ and $Q_k$ (through the path of $Q_i$s) without leaving any cycles or disconnecting the graph. In other words, this gadget behaves like an unbreakable degree-2 vertex, as desired.

Thus we see that the reduction goes through and Graph ($\{k\}, \emptyset$)-TRVB is NP-hard if $k \geq 4$. 

\[ \square \]
**Corollary 8.3.** Graph \((B, U)\)-TRVB is NP-complete if \(B\) contains any \(k \geq 4\).

**Proof.** We saw in Theorem 8.2 that Graph \((\{k\}, \emptyset)\)-TRVB is NP-hard if \(k \geq 4\), and we saw in Lemma 3.2, there is a reduction from Graph \((\{k\}, \emptyset)\)-TRVB to Graph \((B, U)\)-TRVB if \(B\) contains \(k\). Thus, if \(k \geq 4\) and \(B\) contains \(k\), then Graph \((B, U)\)-TRVB is NP-hard. Using Corollary 3.4 we see that as desired, if \(k \geq 4\) and \(B\) contains \(k\), then Graph \((B, U)\)-TRVB is NP-complete. \(\square\)

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

The goal of this section is to show that 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 \(k \in B\). To do this, we will demonstrate that these conditions are sufficient to guarantee that it is possible to build small planar gadgets which behave like unbreakable degree-2 vertices. Inserting two copies of such a gadget into every edge converts a planar multigraph into a planar graph while keeping the answer to the TRVB question the same.

Below, we prove the existence of the desired gadgets.

**Lemma 9.1.** There exists a planar gadget that simulates an unbreakable vertex of degree-2 built out of breakable degree-\(k\) vertices (for any \(k \geq 4\)) and unbreakable degree-4 vertices such that the number of nodes is constant with respect to the size of a given multigraph \(G\).

**Proof.** The gadget for this theorem is shown in Figure 22. For each breakable vertex in this figure, there exists a cycle in the gadget containing the vertex and no other breakable vertex. To avoid leaving this cycle in the final graph, the two breakable vertices in the gadget must both be broken in a valid solution. This fully determines the solution of the gadget.

![Figure 22](image)

Figure 22: 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.

Thus, if this gadget is included in a graph, the two breakable vertices must be broken, resulting in the gadget connecting the two edges that extend out to the rest of the graph. In other words, the gadget behaves like a degree-2 unbreakable vertex.

Note also that this gadget uses \(k + 2\) nodes, which is constant with respect to the size of a given multigraph \(G\). \(\square\)

**Lemma 9.2.** There exists a planar gadget that simulates an unbreakable degree-2 vertex built out of breakable degree-\(k\) vertices (for any \(k \geq 4\)) and unbreakable degree-3 vertices such that the number of nodes is constant with respect to the size of a given multigraph \(G\).
Proof. The gadget for this theorem is shown in Figure 23. The one breakable vertex in this figure is in a cycle in the gadget (with no other breakable vertex). To avoid leaving this cycle in the final graph, the breakable vertex must be broken in a valid solution. This fully determines the solution of the gadget.

Figure 23: 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.

Thus, if this gadget is included in a graph, the breakable vertex must be broken, resulting in the gadget connecting the two edges that extend out to the rest of the graph. In other words, the gadget behaves like a degree-2 unbreakable vertex.

Note also that this gadget uses \(k + 1\) nodes, which is constant with respect to the size of a given multigraph \(G\).

Lemma 9.3. There exists a planar gadget that simulates an unbreakable degree-2 vertex built out of breakable degree-\(k\) vertices (for any \(k \geq 4\)) and unbreakable degree-1 vertices such that the number of nodes is constant with respect to the size of a given multigraph \(G\).

Proof. The gadget for this theorem is shown in Figure 24. The one breakable vertex in this figure cannot be broken (as that would separate the unbreakable vertices from the rest of the graph).

Figure 24: 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.

Thus, if this gadget is included in a graph, the breakable vertex must not be broken, resulting in the gadget connecting the two edges that extend out to the rest of the graph. In other words, the gadget behaves like a degree-2 unbreakable vertex.

Note also that this gadget uses \(k - 1\) nodes, which is constant with respect to the size of a given multigraph \(G\).

Lemma 9.4. There exists a planar gadget that simulates an unbreakable degree-2 vertex built out of breakable degree-\(k\) vertices (for any \(k \geq 4\)) and breakable degree-1 vertices such that the number of nodes is constant with respect to the size of a given multigraph \(G\).

Proof. Breaking a degree-1 vertex does nothing, so breakable degree-1 vertices are essentially the same as unbreakable degree-1 vertices. Thus we can simply use the same construction as for the previous theorem.
Lemma 9.5. There exists a planar gadget that simulates an unbreakable degree-2 vertex built out of breakable degree-$k$ vertices (for any $k \geq 4$) and breakable degree-2 vertices such that the number of nodes is constant with respect to the size of a given multigraph $G$.

Proof. We begin by constructing the gadget shown in Figure 25. In this gadget, breakable vertex $Q$ has $2a$ edges to other vertices $P_0, P_1, \ldots, P_{2a-1}$ in the gadget and $k-2a$ edges out of the gadget. In addition, there is an edge between $P_{2i}$ and $P_{2i+1}$ for every $i$ in $\{0, 1, \ldots, a-1\}$. Note that the degree of $Q$ is $k$ and the degree of each $P_i$ is 2. When solving a graph containing this gadget, the cycle $(Q, P_{2i}, P_{2i+1})$ guarantees that exactly one of the three vertices in the cycle must be broken. $Q$, however, cannot be broken without disconnecting $P_{2i}$ and $P_{2i+1}$ from the rest of the graph. Thus, provided $a > 0$, every valid solution will break exactly one $P_{2i+j}$ out of every pair $(P_{2i}, P_{2i+1})$ and will not break $Q$. If this is done, the part of the resulting graph corresponding to this gadget will connect the $k-2a$ external neighbors of $Q$ to each other without forming any cycles. In other words the behavior of this gadget in a graph is the same as the behavior of an unbreakable degree-$(k-2a)$ vertex.

Figure 25: 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.

Note that the number of nodes in the above gadget is constant with respect to the size of a given multigraph $G$ (in particular, there are $2a+1 \leq k+1$ nodes).

Since $k \geq 4$, we can select $a > 0$ such that $k-2a \in \{2, 3\}$. In other words, the above gadget behaves either as an unbreakable degree-2 vertex gadget or as an unbreakable degree-3 vertex gadget.

If the gadget behaves as an unbreakable degree-3 vertex gadget, then an unbreakable degree-2 vertex gadget can be built (as in the previous theorem) using breakable degree-$k$ vertices and unbreakable degree-3 vertex gadgets. In this case, the size of the new combined gadget is at most a constant times the size of the above gadget.

Thus in all cases we can construct a gadget simulating an unbreakable degree-2 vertex using only degree-$k$ and degree-2 breakable vertices.

Lemma 9.6. There exists a planar gadget that simulates an unbreakable degree-2 vertex built out of breakable degree-3 vertices such that the number of nodes is constant with respect to the size of a given multigraph $G$.

Proof. The gadget for this theorem is shown in Figure 26. If either vertex $P$ or vertex $Q_2$ is broken, then none of the others can be (since all the non-$P$ vertices are adjacent to $P$ and all the non-$Q_2$ vertices are adjacent to $Q_2$). If neither $P$ nor vertex $Q_2$ is broken, then in order to avoid having cycles, both $Q_1$ and $Q_3$ must be broken; this however, disconnects $P$ and $Q_2$ from the rest of the graph. Thus the only valid solutions of this gadget break exactly one of $P$ and $Q_2$ and nothing
else. In either case, the resulting graph piece connects the two edges that extend out to the rest of the graph. In other words, the gadget behaves like a degree-2 unbreakable vertex.

![Diagram](image1)

Figure 26: 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.

Note also that this gadget uses 4 nodes, which is constant with respect to the size of a given multigraph $G$.

**Lemma 9.7.** There exists a planar gadget that simulates an unbreakable degree-2 vertex built out of breakable degree-4 vertices such that the number of nodes is constant with respect to the size of a given multigraph $G$.

**Proof.** The gadget for this theorem is shown in Figure 27. Note that this is actually the same gadget as described in Theorem 8.2 for $k = 4$. Thus we have already argued the correctness of this gadget.

![Diagram](image2)

Figure 27: 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.

Note also that this gadget uses 5 nodes, which is constant with respect to the size of a given multigraph $G$.

**Lemma 9.8.** There exists a planar gadget that simulates an unbreakable degree-2 vertex built out of breakable degree-5 vertices such that the number of nodes is constant with respect to the size of a given multigraph $G$.

**Proof.** The gadget for this theorem is shown in Figure 28.

This gadget contains exactly thirty-two degree-5 vertices with two edges leaving the gadget. A choice of vertices to break within the gadget is a valid solution if either (1) the resulting graph restricted to the vertices within the gadget is a tree (with the two edges extending out of the gadget connected to this tree) or (2) the resulting graph restricted to the vertices within the gadget consists of two trees (with the two edges extending out of the gadget connected to one of these trees each). Since there are 32 degree-5 vertices with two edges out of the gadget, the number of edges in the gadget is $\frac{32 \times 5 - 2}{2} = 79$. Note that breaking vertices does not affect the number of edges in a graph, so the final tree or pair of trees in a valid solution of the gadget will also have 79 edges. A tree with 79 edges has 80 vertices while two trees with 79 edges have 81 vertices. Breaking one vertex increases the number of vertices by 4. Thus, it is possible to achieve the one-tree solution by breaking $\frac{80 - 32}{4} = 12$ vertices and it is impossible to achieve the two-tree solution.
The one-tree solution corresponds with the situation in which the gadget connects the two edges that extend out to the rest of the graph. In other words, provided that the gadget can be solved, every possible solution is one under which the gadget behaves like a degree-2 unbreakable vertex. There is, however, still the question of whether the gadget can be solved. In fact, breaking every vertex above or below the center horizontal line of the gadget (and leaving the 20 vertices along the center line unbroken) is a valid solution of the gadget.

Therefore this gadget behaves like a degree-2 unbreakable vertex.

Note also that this gadget uses 32 nodes, which is constant with respect to the size of a given multigraph $G$.

With these gadgets, we can now reduce from the Planar TRVB variants:

**Theorem 9.9.** 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 $k \in B$.

**Proof.** Suppose that for some $k \geq 4$, we have that $k \in B$ and also that either $B \cap \{1,2,3,4,5\} \neq \emptyset$ or $U \cap \{1,2,3,4\} \neq \emptyset$. Then we can reduce from Planar $(\{k\},\emptyset)$-TRVB to Planar Graph $(B,U)$-TRVB. Our reduction works by inserting either two degree-2 unbreakable vertices or two degree-2 unbreakable vertex gadgets (built out of vertices whose types are allowed in $(B,U)$-TRVB) into each edge. In either case, the resulting multigraph uses only vertices with allowed degrees (with the answer staying the same), but is now also a graph.

There are several cases:

If $B \cap \{3,4,5\} \neq \emptyset$, then let $b$ be an element of $B \cap \{3,4,5\}$. We can build a constant size planar gadget which behaves like an unbreakable degree-2 vertex using only breakable degree-$b$ vertices.
If $B \cap \{1, 2\} \neq \emptyset$, then let $b$ be an element of $B \cap \{1, 2\}$. We can build a constant size planar gadget which behaves like an unbreakable degree-2 vertex using only breakable degree-$b$ vertices and breakable degree-$k$ vertices.

If $U \cap \{1, 3, 4\} \neq \emptyset$, then let $u$ be an element of $U \cap \{1, 3, 4\}$. We can build a constant size planar gadget which behaves like an unbreakable degree-2 vertex using only unbreakable degree-$u$ vertices and breakable degree-$k$ vertices.

If $2 \in U$, then the problem we are reducing to allows degree-2 unbreakable vertices.

Thus, in all cases either (1) the problem we are reducing to allows degree-2 unbreakable vertices or (2) the problem we are reducing to allows vertex types with which we can build a constant sized gadget which behaves like a degree-2 unbreakable vertex.

Thus, our desired reduction from Planar ($\{k\}, \emptyset$)-TRVB to Planar Graph $(B, U)$-TRVB is possible. Since Planar ($\{k\}, \emptyset$)-TRVB is NP-hard (by Corollary \ref{cor:np-hard}), we see that Planar Graph $(B, U)$-TRVB is NP-hard. By Corollary \ref{cor:np-complete}, Planar Graph $(B, U)$-TRVB is in NP, so as desired, Planar Graph $(B, U)$-TRVB is NP-complete.

\begin{acknowledgments}
We would like to thank Zachary Abel and Jayson Lynch for their contributions to this research.
\end{acknowledgments}

\begin{thebibliography}{9}
\item László Lovász. The matroid matching problem. \textit{Algebraic methods in graph theory}, 1:495–517, 1978.
\item Christos H. Papadimitriou and Umesh V. Vazirani. On two geometric problems related to the travelling salesman problem. \textit{Journal of Algorithms}, 5(2):231–246, 1984.
\item J. Plesník. The NP-completeness of the hamiltonian cycle problem in planar digraphs with degree bound two. \textit{Information Processing Letters}, 8(4):199–201, April 1979.
\item Markus Schäffter. Drawing graphs on rectangular grids. \textit{Discrete Applied Mathematics}, 63(1):75–89, 1995.
\item Shuichi Ueno, Yoji Kajitani, and Shin’ya Gotoh. On the nonseparating independent set problem and feedback set problem for graphs with no vertex degree exceeding three. \textit{Discrete Mathematics}, 72(1):355–360, 1988.
\item Hassler Whitney. On the abstract properties of linear dependence. \textit{American Journal of Mathematics}, 57(3):509–533, 1935.
\end{thebibliography}