The Total Interval Number of a Graph, III:
Tree-like Graphs

Thomas M. Kratzke* and Douglas B. West†‡

November 4, 2022

Abstract

A multi-interval representation of a simple graph $G$ assigns each vertex a union of disjoint real intervals so that vertices are adjacent if and only if their assigned sets intersect. The total interval number $I(G)$ is the minimum of the total number of intervals used in such a representation of $G$. We present a linear-time algorithm to compute $I(G)$ when every block of $G$ is a complete graph or a cycle. Also, for an $n$-vertex cactus (every block is an edge or a cycle), the maximum of $I(G)$ is $\lfloor (18n - 12)/13 \rfloor$. For an $n$-vertex block graph (every block is a complete graph), the maximum is $\lfloor 3n/2 - 2 \rfloor$.

1 Introduction

An intersection representation $f$ of a simple graph $G$ assigns each vertex $v$ a set $f(v)$ such that two vertices $u$ and $v$ are adjacent if and only if $f(u) \cap f(v) \neq \emptyset$. A multi-interval representation is an intersection representation in which each $f(v)$ is a union of (closed) intervals on the real line. A $k$-interval is a union of $k$ disjoint real intervals. When $f(v)$ is a $k$-interval, we write $|f(v)| = k$ and say that $v$ is assigned $k$ intervals. Note that $|f(v)|$ is an abuse of notation, since $f(v)$ is an infinite subset of $\mathbb{R}$.

Graphs having multi-interval representations in which each $f(v)$ is a single interval are interval graphs; this class has many applications. To measure how far a graph is from being an interval graph, we may choose $f$ to minimize the maximum or the average of $|f(v)|$ over all vertices. The interval number $i(G)$ of a graph $G$ is the minimum $t$ such that $G$ has a multi-interval representation in which $\max_{v \in V(G)} |f(v)| = t$. The total interval number $I(G)$ is the minimum $t$ such that $G$ has a multi-interval representation in which $\sum_{v \in V(G)} |f(v)| = t$.

*Metron, Inc., Reston, VA 22090, kratzke@metsci.com
†Departments of Mathematics, Zhejiang Normal University and University of Illinois, dwest@illinois.edu
‡This material is based upon work originally supported by the NSA under Award No. MDA904-03-1-0037, with more recent preparation supported by National Natural Science Foundation of China under grants NSFC 11871439, 11971439, and U20A2068.
(dividing by the number of vertices yields the average of \(|f(v)|\)). We write \(|f|\) for \(\sum_v |f(v)|\) and call \(|f|\) the size of \(f\). We allow \(f(v) = \emptyset\), so isolated vertices contribute nothing to \(|f|\).

Interval number was introduced by Trotter and Harary [8] and has been studied in many papers. Although defined in [8] and [3], total interval number was not studied until Aigner and Andreae [1] determined the maximum of \(I(G)\) over several classes of \(n\)-vertex graphs, including trees (\(\lfloor (5n - 3)/4 \rfloor\)), 2-connected outerplanar graphs (\(\lfloor 3n/2 - 1 \rfloor\)), triangle-free planar graphs (\(2n - 3\)), and triangle-free graphs (\(\lceil (n^2 + 1)/4 \rceil\)). For the latter three classes, they conjectured that the upper bounds still hold without the restrictions to 2-connected or triangle-free graphs, respectively.

In [6], we proved these conjectures for outerplanar and general graphs on \(n\) vertices, and we also proved the conjecture from [1] that \(\max I(G) = \lfloor (5m + 2)/4 \rfloor\) for connected graphs with \(m\) edges. The proof that \(I(G) \leq 2n - 3\) for \(n\)-vertex planar graphs is quite lengthy and will appear in the next paper in this series. The final paper will study the maximum of \(I(G)\) for graphs with \(m\) edges satisfying fixed lower bounds on minimum vertex degree, connectivity, or edge-connectivity.

In [7], we provided a linear-time algorithm to compute \(I(G)\) when \(G\) is a tree and observed that recognizing graphs with \(I(G) = |E(G)| + 1\) is NP-complete, where \(|E(G)|\) is the number of edges of \(G\). In this paper, we refine the algorithm for trees to obtain a simpler algorithm that applies more generally. The new algorithm applies to graphs in which every block is a complete graph or a cycle, where a block of a graph is a maximal connected subgraph with no cut-vertex. A cactus is a connected graph in which every block is an edge or a cycle. Hence we call a graph in which every block is a complete graph or a cycle a semi-cactus.

All cacti are outerplanar, but our algorithm as stated does not extend to general outerplanar graphs. Testing \(I(G) = |E(G)| + 1\) is NP-complete even for triangle-free 3-regular planar graphs [7], but the complexity of computing \(I(G)\) for outerplanar graphs is not known.

A block graph is a graph in which every block is a complete graph. Harary [4] showed that a graph has this property if and only if it is the intersection graph of the family of blocks in some graph. The family of semi-cacti contains both all cacti and all block graphs. We determine the maximum value of \(I(G)\) over \(n\)-vertex graphs in these subfamilies. Aigner and Andreae [1] originally suggested that \(I(G) \leq (11n - 4)/8\) when \(G\) is an \(n\)-vertex cactus, but a note added in proof acknowledged the discovery of more extreme cacti in [5].

<table>
<thead>
<tr>
<th>Family of graphs</th>
<th>maximum of (I(G))</th>
</tr>
</thead>
<tbody>
<tr>
<td>(n)-vertex cacti</td>
<td>(\lfloor (18n - 12)/13 \rfloor)</td>
</tr>
<tr>
<td>(n)-vertex block graphs</td>
<td>(\lceil 3n/2 - 2 \rceil)</td>
</tr>
</tbody>
</table>

These extremal results were originally obtained in the first author’s dissertation [5] without the algorithmic approach. Sections 2–5 develop the tools needed to specify the algorithm and prove that it works. In Section 6 we present the algorithm. In Sections 7 and 8 we prove the extremal results for total interval number of \(n\)-vertex cacti and block graphs.
2 Definitions and General Approach

For a graph $G$, we henceforth use representation to mean multi-interval representation and optimal representation to mean a minimum-sized representation. The idea of our algorithm is to obtain an optimal representation of a graph $G$ from optimal representations of certain subgraphs. When $G$ has a cut-vertex $u$, for example, the $u$-lobes of $G$ are the subgraphs consisting of $u$ and a component of $G - u$ together with its edges to $u$. Among the optimal representations of the $u$-lobes, we seek those that use $u$ in helpful ways to allow saving intervals when they are combined to obtain a representation of $G$. For this reason we will keep track of how intervals for $u$ appear in a representation, as described in Definition 2.1.

Combining the representations of $u$-lobes at a cut-vertex $u$ is discussed in Section 3.

Note that when we seek such a good representation of a $u$-lobe, the vertex $u$ is not a cut-vertex. In order to prepare representations of the $u$-lobes to be combined at a cut-vertex, we therefore also need to obtain an optimal representation that uses $u$ in a helpful way when $u$ is not a cut-vertex. In a semi-cactus, such a vertex $u$ lies in a block that is a complete graph or a cycle. The constructions and arguments for inductively obtaining good representations in these cases appear in Sections 4 and 5, respectively.

Together, these three cases yield an inductive algorithm for finding an optimal representation of a semi-cactus with the additional property of having used a specified vertex in a most useful way. This bonus property is analogous to loading the induction hypothesis to facilitate an inductive proof. We next introduce the technical definitions to describe the usefulness of $u$ in a representation.

**Definition 2.1.** For a vertex $v$ in a graph with representation $f$, each boundary point of $f(v)$ is an endpoint (for $v$). The endpoints of a representation $f$ are all points that are endpoints for vertices. We assume that each endpoint of $f$ is an endpoint for exactly one vertex; every representation can be modified slightly to obtain this property without changing its size.

A point $x \in \mathbb{R}$ is displayed in a representation $f$ if belongs to exactly one interval assigned by $f$. An interval is displayed if it contains a displayed point. A vertex $u$ is displayed if $f(u)$ contains a displayed point. An edge $uv$ is displayed if $f(u) \cap f(v)$ contains a point assigned to no other vertex. If $f(u)$ has a displayed endpoint, then $u$ is end-displayed in $f$. We further define the $u$-extent $\epsilon_u(f)$ of a representation $f$ in terms of endpoints and displayed endpoints in $f(u)$, as follows.

<table>
<thead>
<tr>
<th>property of $u$</th>
<th>definition</th>
<th>$\epsilon_u(f)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>non-displayed</td>
<td>$f(u)$ has no displayed point</td>
<td>$-1$</td>
</tr>
<tr>
<td>weakly-displayed</td>
<td>$u$ is displayed but not end-displayed in $f$</td>
<td>$0$</td>
</tr>
<tr>
<td>singly-displayed</td>
<td>$f(u)$ has exactly one displayed endpoint</td>
<td>$1$</td>
</tr>
<tr>
<td>doubly-displayed</td>
<td>$f(u)$ has at least two displayed endpoints</td>
<td>$2$</td>
</tr>
</tbody>
</table>

Note that $\epsilon_u(f) \geq 0$ when $u$ is displayed in $f$ and $\epsilon_u(f) \geq 1$ when $u$ is end-displayed. In a representation $f$, an endpoint of $f(u)$ that is displayed is a $u$-point. The code of a vertex $u$
in $G$, denoted $c(G,u)$, is the maximum of $\epsilon_u(f)$ such that $f$ is an optimal representation of $G$. An optimal representation $f$ of $G$ is $u$-optimal if $\epsilon_u(f) = c(G,u)$. In particular, consider the graph consisting of a single isolated vertex $u$, which is the base case for our recursive algorithm. The total interval number is 0, and there is no displayed interval in an optimal representation, so $c(K_1,u) = -1$.

A component of a representation $f$ is a maximal set of intervals in $f$ whose union is a single interval in $\mathbb{R}$, extending from a displayed left endpoint to a displayed right endpoint. A displayed endpoint is an endpoint of the component containing it.

The intersection graph of a family of multi-intervals is determined by the order of the endpoints of its intervals. In terms of the ordering of endpoints, reflecting a component means reversing the order of its endpoints, and translating a component means moving its sequence of endpoints to a different location in the sequence of components. In this combinatorial description, the geometric operation of shrinking a component has no effect.

When representations of $u$-lobes having $u$-points are combined, it may be possible to reduce the number of intervals used, as illustrated in Figure 1.

\begin{equation*}
\begin{array}{c}
A \quad u \quad B \quad u \\
\quad u \quad A \quad u \\
\end{array}
\begin{array}{c}
\text{splice} \\
\quad B \quad u \quad C
\end{array}
\begin{array}{c}
A \quad u \quad B' \\
\quad B \quad u \quad A \quad u \quad C
\end{array}
\end{equation*}

Figure 1: Splices and swallows

**Definition 2.2.** If $f$ has $u$-points in distinct components, then reflections and translations of components can be used to make these endpoints consecutive. The two intervals can then be extended to meet. This operation is a splice.

If $f$ has two $u$-points in a component $A$ and has another component $B$ in which $u$ is displayed, then $A$ can be inserted into a displayed interval for $u$ in $B$. Geometrically, the interval in $B$ is cut, the component $A$ is inserted, and then two pairs of consecutive endpoints for $u$ are deleted (by extending intervals to meet). This operation is a swallow.

When $u$ is the vertex for which the number of intervals decreases, we call these operations $u$-operations. Each $u$-operation reduces $|f|$, $|f(u)|$, and the number of components by 1 and reduces the number of $u$-points by 2, without changing the graph represented. A representation is $u$-reduced if no $u$-operation is available.

### 3 $u$-Reduction and Cut-vertices

The preceding definitions allow us to describe properties of optimal representations.
Lemma 3.1. An optimal representation \( f \) has at most two \( u \)-points. Furthermore, it can have two \( u \)-points only if they are the endpoints of a single component and no other component has a displayed interval for \( u \).

Proof. A representation having more than two \( u \)-points has more than one component with a \( u \)-point, and then a splice is available. Similarly, two \( u \)-points must belong to the same component, and if \( u \) is displayed in any other component then a swallow is available. \( \square \)

Lemma 3.1 implies that if \( f \) is optimal and \( u \) is displayed in \( f \), then \( \epsilon_u(f) \) is the number of \( u \)-points in \( f \). We use \( u \)-operations in combining \( u \)-optimal representations of graphs to obtain a representation of their union. By Lemma 3.1, optimal representations are \( u \)-reduced for all \( u \in V(G) \).

Definition 3.2. Given graphs \( G_1, \ldots, G_k \), for each \( i \) let \( f_i \) be a representation of \( G_i \). A \( u \)-reduction of \( \{f_i\}_{i=1}^k \) is a representation \( f \) of \( \bigcup G_i \) formed by concatenating \( f_1, \ldots, f_k \) (that is, combining their endpoint sequences end-to-end) and then performing \( u \)-operations until no more \( u \)-operations are available. Note that \( u \) is a single fixed vertex in this process.

A \( u \)-saturated representation is a representation in which displayed points of \( f(u) \) occur only within components of \( f \) whose endpoints both are \( u \)-points.

Lemma 3.3. A representation \( f \) is \( u \)-saturated if and only if in each component, \( u \) is non-displayed or doubly displayed. A \( u \)-reduced representation \( f \) is \( u \)-saturated if and only if \( \epsilon_u(f) \in \{-1, 2\} \). A representation obtained from \( f \) by a \( u \)-operation is \( u \)-saturated if and only if \( f \) is \( u \)-saturated.

Proof. When \( f \) is \( u \)-saturated, the endpoints of any component in which \( u \) has a displayed interval must be \( u \)-points. If \( \epsilon_u(f) \neq 2 \), meaning that \( u \) is not doubly displayed in \( f \), then \( f \) has at most one \( u \)-point, in which case being \( u \)-saturated prevents \( u \) from being displayed. Note that a non-\( u \)-reduced representation \( f \) in which a \( u \)-operation swallow can be performed satisfies \( \epsilon_u(f) = 2 \) but may not be \( u \)-saturated.

When a swallow is performed on \( f \), the component being swallowed satisfies the saturation condition in \( f \), and the receiving component satisfies the condition after the swallow if and only if it satisfies it before the swallow, since at both times it has a displayed interval for \( u \).

When a splice is performed on \( f \), each involved component in \( f \) is guaranteed one \( u \)-point, which is displayed. Hence the saturation condition holds for these components in \( f \) if and only if the other endpoint of each is a \( u \)-point, which holds if and only if the combined component after the splice has two \( u \)-points. \( \square \)

Given \( G = \bigcup_{i=1}^k G_i \) and \( u \)-reduced representations \( f_1, \ldots, f_k \) of \( G_1, \ldots, G_k \), the size and \( u \)-extent of a representation of \( G \) obtained as a \( u \)-reduction of \( f_1, \ldots, f_k \) does not depend on the order of \( u \)-operations performed. We prove this by computing these parameters for the
resulting \( f \) in terms of their values for \( f_1, \ldots, f_k \). We require that \( f_1, \ldots, f_k \) are \( u \)-reduced, but not necessarily \( u \)-optimal.

**Lemma 3.4.** Every \( u \)-reduction \( f \) of \( u \)-reduced representations \( f_1, \ldots, f_k \) has the same size and \( u \)-extent. If \( \beta \) is the total number of \( u \)-points over all \( f_i \), and \( Q \) is the property that all \( f_i \) are \( u \)-saturated, then \( \epsilon_u(f) \) has the value listed below in terms of \( \beta \) and \( Q \). Furthermore, if \( \epsilon_u(f) = 2 \), then \(|f| = \sum |f_i| - \lfloor \beta/2 \rfloor + 1 \); otherwise, \(|f| = \sum |f_i| - \lfloor \beta/2 \rfloor \).

<table>
<thead>
<tr>
<th>( \beta )</th>
<th>( Q )</th>
<th>( \epsilon_u(f) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>true</td>
<td>-1</td>
</tr>
<tr>
<td>even &amp; positive</td>
<td>true</td>
<td>2</td>
</tr>
<tr>
<td>even</td>
<td>false</td>
<td>0</td>
</tr>
<tr>
<td>odd</td>
<td>false</td>
<td>1</td>
</tr>
</tbody>
</table>

**Proof.** Since each \( f_i \) is \( u \)-reduced, by Lemma 3.3 property \( Q \) holds if and only if \( \epsilon_u(f_i) \in \{-1, 2\} \) for all \( i \).

We first consider the case where \( Q \) is true. In this case \( \beta \) is even, since each \( f_i \) contributes 0 or 2 \( u \)-points. The concatenation of \( f_1, \ldots, f_k \) is then \( u \)-saturated, so \( f \) is \( u \)-saturated, by Lemma 3.3. Since \( f \) is \( u \)-reduced, we then have \( \epsilon_u(f) \in \{-1, 2\} \). Since every \( u \)-operation leaves \( u \) displayed, \( \epsilon_u(f) = -1 \) if and only if \( \epsilon_u(f_i) = -1 \) for every \( i \), which requires \( \beta = 0 \) and \(|f| = \sum |f_i| \). When \( \epsilon_u(f) = 2 \), the \( u \)-reduction has two \( u \)-points and thus the number of \( u \)-operations to produce \( f \) is \( \beta/2 - 1 \).

If \( Q \) is false, then some \( f_i \) has a component in which \( u \) is displayed but not doubly displayed. Any \( u \)-operation preserves this property. Hence \( f \) is not \( u \)-saturated. Nevertheless, a \( u \)-operation can be performed as long as there is another \( u \)-point in the representation. Thus the number of \( u \)-operations applied in the reduction is \( \lfloor \beta/2 \rfloor \), and \( \epsilon_u(f) \) is 0 if \( \beta \) is even and 1 if \( \beta \) is odd. \( \square \)

**Corollary 3.5.** Given graphs \( G_1, \ldots, G_k \), for each \( i \) let \( f_i \) and \( f'_i \) be \( u \)-optimal representations of \( G_i \). If \( f \) is a \( u \)-reduction of \( \{f_1, \ldots, f_k\} \) and \( f' \) is a \( u \)-reduction of \( \{f'_1, \ldots, f'_k\} \), then \( f \) and \( f' \) have the same size and \( u \)-extent.

**Proof.** For each \( i \), \( u \)-optimality means that \( f_i \) and \( f'_i \) are \( u \)-reduced and have the same size and \( u \)-extent. Hence a concatenation of \( \{f_1, \ldots, f_k\} \) and a concatenation of \( \{f'_1, \ldots, f'_k\} \) have the same size, number of \( u \)-points, and truth value of \( Q \). Lemma 3.4 then yields \(|f| = |f'|\) and \( \epsilon_u(f) = \epsilon_u(f') \). \( \square \)

The inductive algorithms that we will present combine representations of edge-disjoint subgraphs. Under suitable conditions, \( u \)-optimality of all of \( f_1, \ldots, f_k \) implies \( u \)-optimality of each \( u \)-reduction of \( \{f_1, \ldots, f_k\} \). We reduce the proof of the theorem to the case \( k = 2 \). We begin by applying Lemma 3.4 to the \( u \)-reduction of two \( u \)-reduced representations, enabling us to compute the \( u \)-extent of the \( u \)-reduction from the \( u \)-extents of \( f_1 \) and \( f_2 \).
Corollary 3.6. For a $u$-reduction $f$ of $u$-reduced representations $f_1$ and $f_2$ with $b_i = \epsilon_u(f_i)$,
\[
\begin{align*}
  b_1 + b_2 &\leq 1 \implies |f| = |f_1| + |f_2| \quad \text{and} \quad \epsilon_u(f) = \max\{b_1, b_2\} \\
  b_1 + b_2 &\geq 2 \implies |f| = |f_1| + |f_2| - 1 \quad \text{and} \quad \epsilon_u(f) = b_1 + b_2 - 2
\end{align*}
\]

Proof. In all cases, $-1 \leq b_i \leq 2$. Since each $f_i$ is $u$-reduced, the number of $u$-points in $f_i$ is
max\{0, b_i\}, and $f_i$ is $u$-saturated if and only if $b_i \in \{-1, 2\}$.

The total number of $u$-points is $b_1 + b_2$ unless $\min\{b_1, b_2\} = -1$. If $b_1 + b_2 \leq 1$, then
no $u$-operation occurs (even for $\{b_1, b_2\} = \{-1, 2\}$, since a component in which $u$ is not
displayed cannot swallow a doubly-displayed component). If $b_1 + b_2 \geq 2$, then exactly one
$u$-operation is available (in particular, when $b_1 = b_2 = 2$, each $f_i$ has one component with
two $u$-points, and a single splice or swallow combines them, but then all displayed intervals
for $u$ lie in the component containing the two $u$-points, and no further operation is possible).
This proves the claim for $|f|$.

Lemma 3.4 allows us to compute $\epsilon_u(f)$ when we know the total number of $u$-points in
$f_1$ and $f_2$ ($\beta$ in Lemma 3.4) and the truth of whether $f_1$ and $f_2$ are both $u$-saturated ($Q$
in Lemma 3.4) We have already noted that $\beta = b_1 + b_2$ unless $\min\{b_1, b_2\} = -1$. In that case
$\beta = 0$ if $b_1 = b_2 = -1$ and otherwise $\beta = \max\{b_1, b_2\}$. As noted in the proof of Lemma 3.4,
$Q$ holds if and only if $b_1$ and $b_2$ both lie in $\{-1, 2\}$.

Hence we immediately know $\beta$ and the truth of $Q$ from $\{b_1, b_2\}$. The display below lists
the cases. In each case, we obtain $\epsilon_u(f) = \max\{b_1, b_2\}$ if $b_1 + b_2 \leq 1$ and $\epsilon_u(f) = b_1 + b_2 - 2$
if $b_1 + b_2 \geq 2$. \hfill \box

\[
\begin{array}{|c|c|c|c|}
\hline
\{b_1, b_2\} & b_1 + b_2 & \beta & Q & \epsilon_u(f) \\
\hline
\{2, 2\} & 4 & 4 & \text{true} & 2 \\
\{-1, 2\} & 1 & 2 & \text{true} & 2 \\
\{-1, -1\} & -2 & 0 & \text{true} & -1 \\
\{0, 0\}, \{-1, 0\} & \leq 0 & 0 & \text{false} & 0 \\
\{0, 1\}, \{-1, 1\} & \leq 1 & 1 & \text{false} & 1 \\
\{1, 1\}, \{0, 2\} & 2 & 2 & \text{false} & 0 \\
\{1, 2\} & 3 & 3 & \text{false} & 1 \\
\hline
\end{array}
\]

Theorem 3.7. Let $u$ be a cut-vertex in a graph $G$, with $G_1, \ldots, G_k$ being the $u$-lobes in
$G$. Let $f_1, \ldots, f_k$ be $u$-optimal representations of $G_1, \ldots, G_k$, respectively. Every $u$-reduction
of $\{f_1, \ldots, f_k\}$ is a $u$-optimal representation of $G$. The values of $I(G)$ and $c(G, u)$ are
determined from the numbers $|f_i|$ and $\epsilon_u(f_i)$ for each $i$ as specified in Lemma 3.4, with each
$G_i$ contributing $\max\{0, c(G_i, u)\}$ $u$-points.

Proof. By Corollary 3.5, it suffices to prove that some $u$-optimal representation of $G$ can be
obtained by $u$-reduction of some set of $u$-optimal representations of $G_1, \ldots, G_k$. The proof is
by induction on $k$. The statement is trivial (or vacuous) when $k = 1$. For $k > 2$, we partition
the indices \( \{1, \ldots, k\} \) into nonempty sets \( A_1 \) and \( A_2 \). By Lemma 3.4, the final size and \( u \)-extent of a \( u \)-reduction does not depend on the order of \( u \)-operations. Let \( H_1 = \bigcup_{i \in A_1} G_i \) and \( H_2 = \bigcup_{i \in A_2} G_i \). By the induction hypothesis, \( u \)-reductions of \( \{f_i: i \in A_1\} \) and \( \{f_i: i \in A_2\} \) are \( u \)-optimal for \( H_1 \) and \( H_2 \), respectively. Now the induction hypothesis for \( k = 2 \) guarantees that the final \( u \)-reduction is \( u \)-optimal for \( G \).

Hence it suffices to prove the case \( k = 2 \). Let \( f \) be a \( u \)-reduction of \( u \)-optimal representations \( f_1 \) and \( f_2 \) of \( G_1 \) and \( G_2 \). We will prove that \( f \) has the same size and \( u \)-extent as some \( u \)-optimal representation \( f' \) of \( G \). For an appropriate choice of \( f' \), we will show that \( f' \) is a \( u \)-reduction of \( u \)-reduced representations \( f'_1 \) and \( f'_2 \) for \( G_1 \) and \( G_2 \) (without knowing whether they are \( u \)-optimal). We will then apply Corollary 3.6 to compare \( f' \) and \( f \).

Let \( A = G_1 - u \) and \( B = G_2 - u \). We choose \( f' \) to be a \( u \)-optimal representation of \( G \) minimizing the number of real intervals contained in \( f'(u) \) that connect a component of \( f'(A) \) and a component of \( f'(B) \). We call each such instance a link; note that an assigned interval in \( f(u) \) may contain more than one link. Since \( G \) has no edges joining \( A \) and \( B \), no point in \( f'(u) \) belongs to both \( f'(A) \) and \( f'(B) \), so every link is displayed.

We claim (1) \( f' \) has at most two links, (2) if \( f' \) has two links then they are in the same component and \( f' \) has no \( u \)-point, and (3) if \( f' \) has one link then it has no \( u \)-point outside the component containing the link. Figure 2 illustrates the cases used below to show this, with \( A' \) or \( B' \) indicating reflection of a portion of a representation that doesn’t minimize the number of links.

![Figure 2: Reduction of Links](image)

First suppose that \( f' \) has two links not in the same component of \( f' \). Cutting them, reflecting and/or translating the resulting components, and reattaching the intervals for \( u \) yields a representation with the same size but two fewer links. If \( f' \) has three links in the same component, then cutting the first and third, reflecting the intervening portion of \( f' \), and reattaching again reduces the number of links by two. Hence \( f' \) has at most two links, and if \( f' \) has two links then they are in the same component of \( f' \). If \( f' \) has one link and
a u-point outside the component containing the link, or two links in one component and a u-point anywhere, then cutting a link and performing a splice reduces the number of links by 1 without changing the size.

We now extract from \( f' \) representations \( f'_1 \) and \( f'_2 \) for \( G_1 \) and \( G_2 \). First we cut the links, increasing the number of intervals by 1 for each link, and then every component of the resulting representation establishes edges for \( G_1 \) or \( G_2 \) but not both. Collecting the components that establish edges of \( G_1 \) yields a representation of \( G_1 \). We use this representation as \( f'_1 \) unless it has two components with u-points that arose by cutting two links in one component of \( f' \). In that case, we perform a splice to obtain \( f'_2 \). Any u-operation now available for \( f'_1 \) or \( f'_2 \) would have enabled a u-operation on \( f' \) or a reduction in the number of links in \( f' \). Since \( f' \) was u-reduced and chosen to minimize the number of links, \( f'_1 \) and \( f'_2 \) are now u-reduced.

Furthermore, \( f' \) is a u-reduction of \( \{ f'_1, f'_2 \} \), using just concatenation, one splice, or one swallow, according to whether the number of links in \( f' \) is 0, 1, or 2, respectively.

Let \( \gamma' = |f'_1| + |f'_2| \). We have \( |f'| = \gamma' \) if \( f' \) has no links, and \( |f'| = \gamma' - 1 \) if \( f' \) has one or two links. Since \( f' \) is u-optimal, \( \epsilon_u(f') = c(G, u) \). Let \( b_i = \epsilon_u(f_i) \).

Similarly, let \( \gamma = |f_1| + |f_2| \) and \( c_i = \epsilon_u(f_i) \). Since \( f' \) is a u-reduction of \( \{ f'_1, f'_2 \} \) and \( f \) is a u-reduction of \( \{ f_1, f_2 \} \), Corollary 3.6 yields

\[
\begin{align*}
    b_1 + b_2 &\leq 1 \quad \Rightarrow \quad |f'| = \gamma' \quad \text{and} \quad \epsilon_u(f') = \max\{b_1, b_2\} \\
    b_1 + b_2 &\geq 2 \quad \Rightarrow \quad |f'| = \gamma' - 1 \quad \text{and} \quad \epsilon_u(f') = b_1 + b_2 - 2 \\
    c_1 + c_2 &\leq 1 \quad \Rightarrow \quad |f| = \gamma \quad \text{and} \quad \epsilon_u(f) = \max\{c_1, c_2\} \\
    c_1 + c_2 &\geq 2 \quad \Rightarrow \quad |f| = \gamma - 1 \quad \text{and} \quad \epsilon_u(f) = c_1 + c_2 - 2
\end{align*}
\]

The optimality of \( f' \) yields \( |f| \geq |f'| \), and the optimality of \( f_i \) yields \( |f'_i| \geq |f_i| \) for \( i \in \{1, 2\} \), and thus \( \gamma' \geq \gamma \). We consider three cases. In each case, we show first that \( |f| = |f'| \) and then that \( \epsilon_u(f) \geq \epsilon_u(f') \), which completes the proof that \( f \) is u-optimal.

Case 1: \( c_1 + c_2 \geq 2 \). Here \( |f'| \leq |f| = \gamma - 1 \leq \gamma' - 1 \leq |f'| \). Equality holds throughout, and \( \gamma = \gamma' \) implies \( |f_i| = |f'_i| \). This implies \( c_i \geq b_i \), by the u-optimality of \( f_i \). Now

\[
\epsilon_u(f) = c_1 + c_2 - 2 \geq b_1 + b_2 - 2 = \epsilon_u(f'),
\]

where the last equality follows from \( |f'| = \gamma' - 1 \).

Case 2: \( b_1 + b_2 \leq 1 \). Here \( |f'| = \gamma' \geq \gamma \geq |f| \geq |f'| \). Equality holds throughout, and \( \gamma = \gamma' \) implies \( |f_i| = |f'_i| \). This again implies \( c_i \geq b_i \). Since \( |f| = \gamma \), now

\[
\epsilon_u(f) = \max\{c_1, c_2\} \geq \max\{b_1, b_2\} = \epsilon_u(f'),
\]

Case 3: \( c_1 + c_2 \leq 1 \) and \( b_1 + b_2 \geq 2 \). By symmetry in the indices, we may assume \( b_1 > c_1 \). Since \( \epsilon_u(f'_1) > \epsilon_u(f_1) \), the u-optimality of \( f_1 \) yields \( |f'_1| > |f_1| \). Also \( |f'_2| \geq |f_2| \), by u-optimality of \( f_2 \), so \( \gamma' > \gamma \). Now \( |f'| = \gamma' - 1 \geq \gamma = |f| \geq |f'| \). Again equality holds

9
throughout, so $|f| = |f'|$. Also $|f_1| = |f'_1| - 1$ and $|f_2| = |f'_2|$, so $c_2 \geq b_2$ by the $u$-optimality of $f_2$. Now

$$
\epsilon_u(f) = \max\{c_1, c_2\} \geq c_2 \geq b_2 \geq b_2 + b_1 - 2 = \epsilon_u(f'),
$$

completing the proof that $f$ is $u$-optimal.

\[\square\]

4 Non-cut-vertices in Complete Blocks

Theorem 3.7 allows $u$-optimal representations to be obtained recursively when $u$ is a cut-vertex. To complete an algorithm, we must be able to compute $I(G)$ and $c(G, u)$ recursively when $G$ is connected and $u$ is not a cut-vertex. This section shows how to do that when the block containing $u$ is a complete graph.

In [7], we gave an algorithm for computing $I(G)$ when $G$ is a tree $T$. It computes $c(T, u)$ recursively using the subtrees obtained by deleting $u$. Here we treat a root $u$ of degree $k$ as a cut-vertex and combine representations of the $u$-lobes $T_1, \ldots, T_k$. We then compute $c(T, u)$ from the $c(T_1, u), \ldots, c(T_k, u)$ as described in Section 3. To complete an algorithm that works for trees, here we need only say how to determine $c(T_i, u)$ from $T_i - u$ when $u$ has degree 1 in the $T_i$.

This separates our earlier algorithm into two simpler steps that generalize. The treatment of cut-vertices in Section 3 is valid for all classes of graphs, but recursively computing the code when $u$ is not a cut-vertex requires special properties of the block containing $u$. We will be able to handle such blocks that are complete graphs in this section, cycles in the next.

Definition 4.1. Let $f_1, \ldots, f_k$ be representations of disjoint graphs $G_1, \ldots, G_k$, with $u_i \in V(G_i)$. Let $u$ be a vertex in none of these graphs, and let $U = \{u_1, \ldots, u_k\} \cup \{u\}$. A $U$-completion of $f_1, \ldots, f_k$ is a representation $f$ of the graph $G$ consisting of the disjoint union of $G_1, \ldots, G_k$ plus $u$ and the edges of a complete graph with vertex set $U$, constructed from the concatenation of $f_1, \ldots, f_k$ as follows, with the four constructions illustrated in Figure 3.

1. If some two vertices $u_i$ and $u_j$ among $u_1, \ldots, u_k$ are end-displayed in the representations $f_i$ and $f_j$, then for one such pair extend the end-displayed intervals for $u_i$ and $u_j$ to intersect (after translating and/or reflecting the components containing them, if needed), and add pairwise-intersecting intervals for the rest of $U$ in their intersection.

2. If only one vertex $u_i$ in $u_1, \ldots, u_k$ is end-displayed in the corresponding representation $f_i$, then add intervals for the rest of $U$ containing that endpoint (intersecting no other intervals) and extend the interval for $u$ to be end-displayed.

3. If no vertex among $u_1, \ldots, u_k$ is end-displayed in the corresponding representation but some $u_i$ is displayed in $f_i$, then add intervals for the rest of $U$ containing a displayed point in $f_i(u_i)$ (intersecting no other intervals).
(4) If no vertex among $u_1, \ldots, u_k$ is displayed in the corresponding representation, then form a new component with pairwise-intersecting intervals for all of $U$, extending the interval for $u$ in both directions to be doubly-end-displayed.

Note that the definition of $U$-completion allows the case where some $G_i$ consists only of the vertex $u_i$, in which case $|f_i| = 0$ and $\epsilon_{u_i}(f_i) = -1$.

**Figure 3: $U$-completions**

**Lemma 4.2.** If $f$ is a $U$-completion of representations $f_1, \ldots, f_k$ with $c_i = \epsilon_{u_i}(f_i)$, then $|f|$ and $\epsilon_u(f)$ are determined as follows, where $l$ is the number of intervals other than $u$ that have displayed intervals used in constructing the $U$-completion.

| case       | $c_1, \ldots, c_k$ | $|f| - \sum |f_i|$ | $l$ | $\epsilon_u(f)$ |
|------------|--------------------|------------------|-----|-----------------|
| (1)        | at least two of $c_1, \ldots, c_k$ are positive | $k - 1$ | 2 | -1 |
| (2)        | exactly one of $c_1, \ldots, c_k$ is positive | $k$ | 1 | 1 |
| (3)        | $\max_i c_i = 0$ | $k$ | 1 | -1 |
| (4)        | $\max_i c_i = -1$ | $k + 1$ | 0 | 2 |

**Proof.** The conditions for the four types of $U$-completion described in Definition 3 are precisely described by the conditions on $c_1, \ldots, c_k$ listed above. In each case, the number of intervals added in forming the $U$-completion, plus the number $l$ of previously existing intervals that are used, equals $k + 1$, which is the number of pairwise-intersecting intervals establishing the edges induced by $U$ in the resulting representation. The final column describes the usage of $u$. □

Lemma 4.2 implies that every $U$-completion of $f_1, \ldots, f_k$ has the same size and $u$-extent.

**Definition 4.3.** A clique in a graph is a set of pairwise adjacent vertices. In a representation $f$ for a graph, a pile for a clique $U$ is a maximal interval $I \subseteq \mathbb{R}$ such that (1) only vertices of $U$ are assigned points of $I$, (2) at least two vertices of $U$ are assigned points of $I$, (3) the vertices assigned points of $I$ are assigned a common point in $I$, and (4) among the intervals intersecting $I$, at most one extends to the left of $I$ and at most one extends to the right. The size of a pile $I$ is the number of vertices assigned points of $I$, and its base is the vertex or pair of vertices whose intervals intersecting $I$ extend to the left or right of $I$ (if some such vertex exists). Figure 3 shows four piles for a clique $U$. Each has size 4, the first two have two base vertices, and the last two have one base vertex. We abuse terminology and say that a pile $I$ contains the vertices assigned intervals in it.
Lemma 4.4. Let $u$ be a non-cut-vertex in a graph $G$. If the block containing $u$ is a complete graph with vertex set $U$, then $G$ has a $u$-optimal representation with only one pile for $U$.

Proof. Let $f$ be a representation of $G$ having more than one pile for $U$. It suffices to show that in every case we can obtain a representation $f'$ that is better: either (1) $|f'| < |f|$, or (2) $|f'| = |f|$ and $\epsilon_u(f') > \epsilon_u(f)$, or (3) $|f'| = |f|$ and $\epsilon_u(f') = \epsilon_u(f)$ but $f'$ has fewer piles for $U$ than $f$. Note that inserting or deleting non-displayed intervals does not affect the extent for any vertex.

If some pile contains all of $U$, then we can delete the non-base intervals from other piles and shrink the intersections of the bases to obtain a representation with one pile for $U$. This does not increase the size of the representation or decrease the $u$-extent.

Let $m = |U|$. When $m = 2$, every pile contains $U$. When $m = 3$, let $U - \{u\} = \{v, w\}$. If no pile contains $U$, then $uv$, $uw$, and $vw$ occur in distinct piles. Since the vertex set of the block is $U$, the vertices $v$ and $w$ have no common neighbor other than $u$. Hence the edge $vw$ is displayed. Also, $u$ has no neighbor outside $\{v, w\}$. We can therefore add an interval for $u$ inside $f(v) \cap f(w)$ and delete the other two intervals for $u$ to reduce the size of the representation. Thus we may assume that $m \geq 4$ and that no pile contains $U$.

If the entire base of a pile $P$ is an interval for some vertex $x$, then $x$ appears in another pile $P'$, since no pile contains $U$. The other intervals in $P$ intersect only intervals for vertices of $U$ and can be moved into $P'$, where they again intersect $f(x)$. This reduces the number of piles without changing the number of intervals or the extent of any vertex. Thus we may assume that every pile has intervals for two vertices in its base.

We next claim that the complete subgraphs represented by the piles are edge-disjoint. Suppose that $P$ and $P'$ are two piles having intervals for both $y$ and $z$ in $U$. Let $I'$ be the interval where $P'$ occurs. Delete $f(v) \cap I'$ from $f(v)$ for each $v \in U$ such that $f(v) \cap I' \neq \emptyset$. Add an interval for each such $v$ to the pile $P$ if it is not already there. All edges that were represented in $P$ or $P'$ before are now represented in the augmented pile $P$. We may have added up to two intervals for the base vertices of $P'$ in the augmented pile $P$ (since such intervals involved in $P'$ were shortened but not eliminated), but we eliminated at least two intervals for $\{y, z\}$. Hence we did not increase the total number of intervals. Also we did not reduce the $u$-extent (it may have increased if $u$ is in the base of $P'$). Hence the new representation is as good as the old one, and we have reduced the number of piles.

We next prove that each vertex of $U$ appears in at most two bases. If $v \in U$ appears in three bases, then we may take it to be the left base point twice, with $x$ and $y$ being the right base points in those piles (see Figure 4). Let $A$ and $B$ be the sets of non-base intervals in the two piles, respectively, which may or may not be in the same component of $f$. By symmetry, we may take the pile with $A$ and $x$ to be leftward of the pile with $B$ and $y$. Reflect the portion of $f$ between the two piles (including the intervals for $x$ and $v$ that extend to $A$ and $B$), extend the interval for $x$ now at the right to meet the interval for $y$, and extend the
interval for \( v \) now at the left to merge with the interval for \( v \) that contained the intervals in \( A \). On the base formed by the intervals for \( x \) and \( y \) pile intervals for \( A \cup B \cup \{ v \} \). The total number of intervals did not increase, since we combined two intervals for \( v \) and added one. The number of \( u \)-points is the same as before (\( u \) may be any of \( v, x, y \) or in \( A \cup B \)), but we reduced the number of piles.

\[
\begin{array}{ccc}
A & C & B \\
v & v & y \\
x & \rightarrow & x
\end{array}
\]

Figure 4: A vertex \( v \) in three bases

Since no pile contains \( U \), each vertex of \( U \) appears in at least two piles. Suppose that some vertex \( v \) appears in only two piles, \( P \) and \( P' \). Since piles represent edge-disjoint complete subgraphs, \( P \) and \( P' \) share only \( v \). Since no other pile contains \( v \), and the subgraphs are edge-disjoint, every vertex of \( U - \{ v \} \) appears in exactly one of \( P \) and \( P' \). Since \( m \geq 4 \), by the pigeonhole principle and symmetry we may assume that \( P \) has two vertices \( x \) and \( y \) other than \( v \). To maintain edge-disjointness, every pile other than \( P \) and \( P' \) consists of one interval from \( P \) and one from \( P' \), both being base intervals. Since we have shown that every pile has two base vertices, we may let \( z \) be a base vertex of \( P' \) other than \( v \). Now \( z \) appears as a base vertex in distinct piles with each of \( v, x, y \). This contradicts our previous restriction that each vertex of \( U \) appears in at most two bases.

We are left only with the case where each element of \( U \) appears in at least three piles. Since each element of \( U \) appears in at most two bases, the piles contain at least \( 3m - 2m \) non-base intervals. We now delete all the non-base intervals (at least \( m \) of them) and pile \( U \) on a single still-existing base (adding \( m - 2 \) intervals) to reduce the size of the representation.

**Theorem 4.5.** Let \( G \) be a connected graph having a non-cut-vertex \( u \) in a block \( Q \) that is a complete graph with vertex set \( U \). Let \( G_1, \ldots, G_k \) be the components of \( G - u - E(Q) \), and let \( u_i \) be the vertex of \( G_i \) in \( U \). For \( 1 \leq i \leq k \), let \( f_i \) be a \( u_i \)-optimal representation of \( G_i \). Every \( U \)-completion of \( f_1, \ldots, f_k \) is a \( u \)-optimal representation of \( G \).

**Proof.** Let \( f \) be a \( U \)-completion of \( f_1, \ldots, f_k \); by construction, \( f \) has only one pile for \( U \). Let \( f' \) be a \( u \)-optimal representation of \( G \) having one pile for \( U \), which exists by Lemma 4.4. We follow the method of Theorem 3.7, showing that \( f \) has the same size and \( u \)-extent as \( f' \), which makes \( f \) \( u \)-optimal.

We begin by showing that \( f' \) is a \( U \)-completion of representations \( f'_1, \ldots, f'_k \) of \( G_1, \ldots, G_k \) that we obtain from \( f' \). Since \( f' \) has only one pile for \( U \), we may assume that \( f'(u) \) is a single interval in this pile. Form \( f'_1, \ldots, f'_k \) by deleting the non-base intervals of the pile, shrinking the intervals of the base to separate them (if the base has two intervals), deleting the interval for \( u \) if it was part of the base (this happens if and only if \( e_u(f') \geq 1 \)), and letting \( f'_i \) consist
of the components of the resulting representation that contain intervals for vertices of $G_i$. The operation that returns to $f'$ from $f'_1, \ldots, f'_k$ is $U$-completion.

As in Theorem 3.7, let $\gamma = \sum |f_i|$, $c_i = \epsilon_u(f_i)$, $\gamma' = \sum |f'_i|$, and $b_i = \epsilon_u(f'_i)$. Note that $c_i = c(G_i, u)$, but we know nothing about $b_i$, since we know nothing about optimality of $f'_i$.

Since $f$ and $f'$ are $U$-completions, we can apply Lemma 4.2 to both. Define $l$ and $l'$ by $|f| - \gamma = k + 1 - l$ and $|f'| - \gamma = k + 1 - l'$. By Lemma 4.2, $l$ and $l'$ are the numbers of vertices other than $u$ in the base of the piles for $U$ in $f$ and $f'$ respectively. In particular, $l, l' \in \{0, 1, 2\}$. The relationship between $l$ and $c_1, \ldots, c_k$ is given by Lemma 4.2, as is the relationship between $l'$ and $b_1, \ldots, b_k$, since $f$ and $f'$ are $U$-completions.

Optimality yields $|f| \geq |f'|$ and $|f'_i| \geq |f_i|$, and the latter implies $\gamma' \geq \gamma$. Consider first the case $l' \leq l$. We compute

$$|f'| \leq |f| = \gamma + (k + 1) - l \leq \gamma' + (k + 1) - l' = |f'|.$$

Equality must hold throughout, and thus $|f| = |f'|$, $\gamma' = \gamma$, and $l = l'$. If $f'$ is $u$-optimal but $f$ is not, then we must have $\epsilon_u(f) < \epsilon_u(f')$. By Lemma 4.2, the only flexibility in the $u$-extent in terms of the common value of $l$ and $l'$ is when that value is 1. Hence $\epsilon_u(f) = -1$ and $\epsilon_u(f') = 1$. Furthermore, we find from Lemma 4.2 that max $c_i = 0$ and max $b_i > 0$ (exactly one of $b_1, \ldots, b_k$ is positive). However, since $\gamma' = \gamma$ and each $f_i$ is $u_i$-optimal, we have $|f'_i| = |f_i|$ for each $i$, and hence $u_i$-optimality of $f_i$ requires $c_i \geq b_i$. We have obtained a contradiction, and hence $f$ must be $u$-optimal.

In the remaining case, $l' > l$. Since $l, l' \in \{0, 1, 2\}$, this requires $l' = 2$ or $l = 0$. If again $\gamma' = \gamma$, then again $|f'_i| = |f_i|$ and $c_i \geq b_i$ for all $i$. However, $l' > 0 = l$ requires max $b_i \geq 0 >= -1 = \max c_i$, and $l' = 2 > l$ requires that at least two of $\{b_i\}$ and at most one of $\{c_i\}$ are positive. Both cases yield some $j$ such that $b_j > c_j$. Thus $l' > l$ implies $\gamma' > \gamma$. In particular, $\gamma \leq \gamma' - 1$.

If $l' = l + 1$, then

$$|f'| \leq |f| = \gamma + (k + 1) - l \leq (\gamma' - 1) + (k + 1) - l' + 1 = |f'|.$$

Again equality holds throughout, and $|f| = |f'|$. If $l = 0$, then $\epsilon_u(f) = 2$, while $l' = 2$ yields $\epsilon_u(f') = -1$, by Lemma 4.2. Thus $\epsilon_u(f) \geq \epsilon_u(f')$, and $f$ is $u$-optimal.

Finally, suppose $l' = 2$ and $l = 0$. By Lemma 4.2, $\epsilon_u(f) = 2 > -1 = \epsilon_u(f')$. Since $f'$ is $u$-optimal, this requires $|f| \geq |f'| + 1$. Also $|f| - \gamma = k + 1 = |f' - \gamma' + 2$, so

$$|f'| - \gamma' + 2 = |f| - \gamma \geq |f'| + 1 - \gamma,$$

which simplifies to $\gamma \geq \gamma' - 1$. We earlier proved $\gamma \leq \gamma' - 1$ when $l' > l$, so equality holds. With $\gamma' = \gamma + 1$, we have $|f'_i| = |f_i|$ and thus $c_i \geq b_i$ for all but one value of $i$. However, $l' = 2$ requires two positive values in $\{b_1, \ldots, b_k\}$, while $l = 0$ requires max $c_i = -1$.

Given that $f'$ is $u$-optimal, we have obtained contradictions in all cases unless $|f| = |f'|$ and $\epsilon_u(f) = \epsilon_u(f')$, so we conclude that $f$ is $u$-optimal. 

\[\Box\]
5 Non-cut-vertices in Cycle Blocks

In order to complete the algorithm for semi-cacti, we must also construct a \( u \)-optimal representation when \( u \) is a non-cut-vertex whose block is a chordless cycle; the case of a 3-cycle is handled as a clique block. We begin by introducing notation for the case of cycle blocks.

**Definition 5.1.** Let \( G \) be a graph having a block that is a cycle \( C \) with vertices \( u, u_1, \ldots, u_k \) in order, where \( k \geq 3 \). The neighbors of vertex \( u \) are \( u_1 \) and \( u_k \). Deleting \( u \) and the edges of \( C \) leaves disjoint graphs \( G_1, \ldots, G_k \), with \( u_i \in V(G_i) \).

A small interval in a representation is an interval whose endpoints are consecutive in the ordering of all endpoints. An insertion is a small interval inside another interval, with the pair representing an edge that does not lie in a triangle. A hook in a representation is a pair of overlapping intervals representing an edge that does not lie in a triangle.

Henceforth in this section, \( G \) is as described in Definition 5.1. We will construct a \( u \)-optimal representation of \( G \) from \( f_1, \ldots, f_k \), where \( f_i \) is a \( u_i \)-optimal representation of \( G_i \).

Our first step is to reduce the problem to making use of such \( f_1, \ldots, f_k \) and representing the edges of \( C \), taking advantage of displayed endpoints for \( u_1, \ldots, u_k \) to minimize the number of added intervals.

**Lemma 5.2.** Given a graph \( H \), let \( F \) be a set of edges that lie in no triangle, and fix \( v \in V(H) \). There is a \( v \)-optimal representation of \( H \) in which each edge of \( F \) is represented irredundantly. Also, a representation of \( H - F \) can be obtained by iteratively, for each \( e \in F \), removing one interval for an endpoint of \( e \) or shrinking intervals for the endpoints of \( e \) to eliminate the intersection.

**Proof.** Consider \( e \in F \) with endpoints \( x \) and \( y \). Since \( e \) lies in no triangle, in any representation \( f \) the intersection of \( f(x) \) and \( f(y) \) does not intersect \( f(z) \) for any \( z \in V(H) - \{x, y\} \). The edge \( e \) is then represented by two intervals forming an insertion or a hook. In the first case, one can delete the small interval; in the second, one can delete the intersection or simply shrink the intervals involved to eliminate the intersection. Neither operation increases the number of intervals or reduces the code for any vertex. If \( e \) had been redundantly represented, then the operation reduces the number of redundantly represented edges; otherwise, a representation of \( H - e \) is produced.

If we began with a \( v \)-optimal representation and applied the operation first to redundantly represented edges, then we obtained the desired \( v \)-optimal representation of \( H \). Continuing, we also verify the second claim. \( \square \)

**Definition 5.3.** Let \( f \) be a representation of a graph \( G \) as described in Definition 5.1, with \( k \geq 3 \). The \( C \)-extraction from \( f \) consists of representations \( f_1, \ldots, f_k \) of \( G_1, \ldots, G_k \), respectively, obtained by deleting the edges of \( C \) as in Lemma 5.2, collecting as \( f_i \) the
components of the resulting representation of \(G - E(C)\) that represent edges in \(G_i\), deleting isolated intervals, and performing one \(u_i\)-splice for each \(i\) such that \(c(u_i, G_i) = 0\) and such a splice is available.

**Lemma 5.4.** Let \(u\) be a non-cut-vertex in a cycle block \(C\) in a graph \(G\). The graph \(G\) has a \(u\)-optimal representation \(f\) whose \(C\)-extraction consists of a \(u_i\)-optimal representation of \(G_i\) for each \(i\) with \(1 \leq i \leq k\).

**Proof.** Let \(f'\) be a \(u\)-optimal representation of \(G\). We may assume by Lemma 5.2 that the edges of \(C\) are represented irredudantly in \(f'\). Let \(f'_1, \ldots, f'_k\) be the \(C\)-extraction from \(f'\). For each \(i\), let \(f_i\) be a \(u_i\)-optimal representation of \(G_i\).

We will replace \(f'\) by a representation \(f\) whose \(C\)-extraction replaces \(f'_i\) with \(f_i\), without increasing the number of intervals or decreasing the \(u\)-extent if it leaves the size unchanged. Repeating this for all \(i\) produces the desired representation. Note that \(u_i\)-optimality of \(f_i\) implies \(|f'_i| \geq |f_i|\) and, if equality holds, then \(\epsilon_{u_i}(f_i) \geq \epsilon_{u_i}(f'_i)\).

Let \(u_0 = u = u_{k+1}\). Now let \(x = u_{i-1}\) and \(y = u_{i+1}\); note that \(x\) or \(y\) may be \(u\). Let \(J_x\) and \(J_y\) be the intervals for \(x\) and \(y\) whose intersection represents the edge \(xu_i\). Similarly, let \(J_i^-\) and \(J_i^+\) be the intervals for \(u_i\) and \(y\) whose intersection represents the edge \(u_iy\). The edges \(xu_i\) and \(u_iy\) may be represented by an insertion or by a hook. Because \(x\) or \(y\) may be \(u\), we must take care that replacing \(f'_i\) by \(f_i\) does not decrease the \(u\)-extent.

Suppose first that \(xu_i\) and \(u_iy\) are both represented by insertions. If \(J_i^-\) and \(J_i^+\) are inserted into \(J_x\) and \(J_y\), then the extraction deletes \(J_i^-\) and \(J_i^+\), leaving \(f'_i\). In particular, the components of \(f'_i\) are not modified in forming \(f'\). Hence we can replace \(f'_i\) with \(f_i\) to form \(f\), with no increase in intervals or decrease in \(u\)-extent. (Note that when \(u \in \{x, y\}\), the possibility of deleting \(J_x\) or \(J_y\) to finish the extraction does not affect the \(u\)-extent in the full representation.)

If \(J_x\) and \(J_y\) are inserted into \(J_i^-\) and \(J_i^+\), then we can replace \(f'_i\) with \(f_i\) and perform the insertions into a displayed portion of \(f_i(u_i)\) (without decreasing the \(u\)-extent), unless \(\epsilon_{u_i}(f_i) = -1\). However, the insertions into \(J_i^-\) and \(J_i^+\) in \(f'\) imply \(\epsilon_{u_i}(f'_i) \geq 0\). Since \(f_i\) is \(u_i\)-optimal, we have \(\epsilon_{u_i}(f_i) \geq 0\) unless \(|f_i| < |f'_i|\). In that case we replace \(f'_i\) with \(f_i\) and have the freedom to add a doubly end-displayed interval for \(u_i\) containing small intervals \(J_x\) and \(J_y\). Now we obtain \(f_i\) in the \(C\)-extraction of the new representation of \(G\).

If exactly one of \(J_i^-\) and \(J_i^+\) is a small interval, then by symmetry we may assume that \(J_i^-\) is inserted into \(J_x\) and \(J_y\) is inserted into \(J_i^+\). In this case we replace \(J_i^-\) by a new small interval for \(x\) inserted into \(J_i^+\), calling that \(J_x\) and letting \(J_i^+ = J_i^-\). The size does not change. Also, the \(u\)-extent does not change, even if \(x = u\), because the intervals for \(x\) that were originally present remain unchanged. We have now produced exactly the case in the previous paragraph.

Next suppose that exactly one of \(xu_i\) and \(u_iy\) is represented in \(f'\) by a hook; by symmetry, we may assume it is \(xu_i\). If \(J_i^-\) does not disappear in the extraction, then \(\epsilon_{u_i}(f'_i) \geq 1\). Now
either \( \epsilon_u(f_i) \geq 1 \) or \(|f_i| < |f'_i|\). We replace \( f'_i \) with \( f_i \) and can either extend an end-displayed interval for \( u_i \) to form the hook or have the freedom to add an interval for \( u_i \) as \( J^-_i \) to use in the hook. The insertion of \( J^+_i \) into \( J_y \) or \( J_y \) into \( J^+_i \) can be replaced by inserting a small interval for \( y \) into \( J^-_i \). As in the previous case, the original intervals for \( y \) remain unchanged (except possibly for moving a small interval), so the \( u \)-extent remains unchanged even if \( y = u \). Now \( f_i \) is in the \( C \)-extraction of the new representation of \( G \), and we have not increased the size or reduced the \( u \)-extent.

Hence we may assume that both \( xu_i \) and \( u_i y \) are represented by hooks. If \( J^-_i \) and \( J^+_i \) lie in the same component of \( f'_i \), then \( \epsilon_u(f'_i) = 2 \). By the \( u \)-optimality of \( f_i \), either \( \epsilon_u(f_i) = 2 \) or \(|f_i| < |f'_i|\). As in the previous paragraph, we can replace \( f'_i \) by \( f_i \), adding an interval for \( u_i \) if \(|f_i| < |f'_i|\), and form the two hooks using the available endpoints for \( u_i \) as they were formed in \( f' \).

Now suppose that \( J^-_i \) and \( J^+_i \) lie in different components of \( f'_i \). We can apply a \( u \)-splice to \( f'_i \) to obtain a representation \( g_i \) of \( G_i \) using \(|f'_i| + 1\) intervals, with \( \epsilon_u(g_i) \geq 0 \). Hence \(|f'_i| > |f_i|\), since \( f_i \) is \( u \)-optimal. If \( \epsilon_u(f_i) \geq 1 \), then replacing \( f'_i \) with \( f_i \) and adding an interval for \( u_i \) again allows us to form \( f \) with \( xu_i \) and \( u_i y \) both being represented by hooks and \( \{J^-_i, J^+_i, J_x, J_y\} \) end-displayed if they were end-displayed in \( f' \). If \( \epsilon_u(f_i) = -1 \), then \( \epsilon_u(g_i) \geq 0 \) and \( u \)-optimality of \( f_i \) require \(|f'_i| \geq |g_i| + 1 \geq |f_i| + 2 \). Now we have the freedom to add two intervals for \( u_i \) to create the hooks when we replace \( f'_i \) by \( f_i \) in \( f' \).

Again given \( g_i \), consider the case \( \epsilon_u(f_i) = 0 \). Now \(|f'_i| - 1 = |g_i| \geq |f_i|\), so we have the freedom to increase the number of intervals for \( u_i \) by 1 when we replace \( f'_i \) by \( f_i \). To do this, split a displayed interval for \( u_i \) in \( f_i \) into two intervals by inserting two endpoints. This turns one component into two components having \( u_i \)-points, increasing the number of intervals by 1. Translations and reversals of these two components as needed allow us to recreate the hooks. The resulting representation \( f \) has no more intervals than \( f' \), and its \( C \)-extraction (after the available \( u \)-splice) includes the \( u \)-optimal \( f_i \) for \( G_i \).

Now that we may assume that the representations \( f_i \) in the \( C \)-extraction are \( u \)-optimal, finding what needs to be added to obtain a \( u \)-optimal representation of \( G \) will depend only on the codes of \( G_1, \ldots, G_k \).

**Definition 5.5.** Given \( G \) as in Definition 5.1, the **codelist** \( c(G) \) of \( G \) is the \( k \)-tuple \((c_1, \ldots, c_k)\) defined by \( c_i = c(u_i, G_i) \) for \( 1 \leq i \leq k \). The **canonical instance** of \( G_i \) for each value of \( c_i \) is as follows.

<table>
<thead>
<tr>
<th>( c_i )</th>
<th>( G_i )</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>( P_1 ) (no edge)</td>
</tr>
<tr>
<td>0</td>
<td>( P_5 ) with ( u_i ) as the central vertex</td>
</tr>
<tr>
<td>1</td>
<td>( P_3 ) with ( u_i ) as an endpoint</td>
</tr>
<tr>
<td>2</td>
<td>( P_2 ) with ( u_i ) as an endpoint</td>
</tr>
</tbody>
</table>

17
For each canonical instance, the total interval number is the number of vertices, and \( c(u_i, G_i) \) is as listed.

**Lemma 5.6.** Given \( G \) as in Definition 5.1, let \( f \) be a \( u \)-optimal representation of \( G \), and let \( f_1, \ldots, f_k \) be the \( C \)-extraction from \( f \). Let \( f' \) be a \( u \)-optimal representation of the graph \( G' \) obtained by replacing each \( G_i \) with the corresponding canonical instance \( G'_i \) such that \( c(u_i, G'_i) = c(u_i, G_i) \). Let \( \gamma_i = I(G_i) \) and \( \gamma'_i = I(G'_i) = |V(G'_i)| \). If \( \gamma = \sum \gamma_i \) and \( \gamma' = \sum \gamma'_i \), then \( I(G) - \gamma = I(G') - \gamma' \), and \( c(u, G) = c(u, G') \).

**Proof.** By Lemma 5.4, we may assume that each \( f_i \) is a \( u_i \)-optimal representation of \( G_i \). Note also that a \( u_i \)-optimal representation \( f'_i \) of \( G'_i \) is as described in Definition 5.5, and by Lemma 5.4 we may assume that these are used in a \( u \)-optimal representation of \( G' \). Indeed, these are the only \( u_i \)-optimal representations for \( G'_1, \ldots, G'_k \).

From \( f_i \) only intervals for \( u_i \) are used in representing edges of \( C \). In the \( u_i \)-optimal representation \( f_i \), intervals for \( u_i \) are available as described by \( c_i \). Since \( G'_i \) has the same code as \( G_i \), in \( f'_i \) the intervals for \( u_i \) are available in the same way. Hence whatever we do with \( f_1, \ldots, f_k \) to produce a \( u \)-optimal representation of \( G \) can also be done with \( f'_1, \ldots, f'_k \) to produce a \( u \)-optimal representation of \( G' \), and conversely. Hence the number of intervals added in both cases are the same, and similarly \( c(u, G) = c(u, G') \). \( \square \)

Having reduced \( G \) to a canonical graph with the same codelist, we further modify the problem by simplifying the codelist. We will consider various transformations, which we call reductions. The first reduces the problem to instances whose codelist contains no 0. The others all reduce the length of the codelist. However, we can only apply these reductions when the resulting codelist has length at least 3, leaving us with a finite number of small graphs as a basis. The cost of the reduction is the amount by which it reduces \( I(G) - \gamma \); we must add this many extra intervals to obtain the \( u \)-optimal representation of the original graph from that of its \( C \)-extraction.

**Definition 5.7. Codelist reductions.**

<table>
<thead>
<tr>
<th>codelist change</th>
<th>restriction</th>
<th>cost in intervals</th>
</tr>
</thead>
<tbody>
<tr>
<td>((0) \rightarrow (1,1))</td>
<td>(k \geq 4)</td>
<td>0</td>
</tr>
<tr>
<td>((-1,-1) \rightarrow (2))</td>
<td>(k \geq 4)</td>
<td>2</td>
</tr>
<tr>
<td>((2,-1)) or ((-1,2) \rightarrow (2))</td>
<td>(k \geq 4)</td>
<td>1</td>
</tr>
<tr>
<td>((1,-1,2) \rightarrow (1,1))</td>
<td>(k \geq 4)</td>
<td>1</td>
</tr>
<tr>
<td>((1,1,1) \rightarrow (1))</td>
<td>(k \geq 5)</td>
<td>1</td>
</tr>
<tr>
<td>((2,1,1,2) \rightarrow (2,2,2))</td>
<td>(k \geq 4)</td>
<td>1</td>
</tr>
<tr>
<td>((2,1,2) \rightarrow (2,2))</td>
<td>(k \geq 4)</td>
<td>1</td>
</tr>
<tr>
<td>((2,2) \rightarrow (2))</td>
<td>(k \geq 4)</td>
<td>0</td>
</tr>
</tbody>
</table>

After eliminating 0s, the next three reductions eliminate all \(-1\)s from the codelist except possibly an isolated \(-1\) at the beginning or end. Reduction 4 then shortens segments of 1s
to length at most 2, after which we can eliminate singleton or doubleton 1s caught between 2s. Note that maintaining \( k \geq 3 \) leaves some ambiguity for the final base list. For example, when the list is \((2,1,1,2,1,2)\), we can apply reduction 5 to eventually finish at \((2,1,2)\) or reduction 6 to eventually finish at \((2,1,1,2)\).

The next eight lemmas justify the reductions in Definition 5.7, including the cost of reversing the reduction. The proofs follow the same template.

**Remark 5.8. Template for validation of reductions.** We begin with \( G \) as described in Definition 5.1, with cycle \( C \) the deletion of whose edges leaves \( u \) and \( G_1, \ldots, G_k \) having codelist \((c_1, \ldots, c_k)\). First we invoke Lemma 5.4 to ensure that the \( C \)-extraction of a \( u \)-optimal representation of \( G \) will consist of \( u_i \)-optimal representations \( f_1, \ldots, f_k \) of \( G_1, \ldots, G_k \). Next, Lemma 5.6 allows us to assume, with appropriate adjustment in the total number of intervals, that each \( G_i \) is the canonical instance as listed in Definition 5.5.

Let \( G' \) be the graph with cycle \( C' \) and canonical instances giving codelist \( c' \) that results from the specified reduction as listed in Definition 5.7. Again the canonical instances and the \( C' \)-extraction \( f'_1, \ldots, f'_{k'} \) result from Lemmas 5.4 and 5.6.

Let \( \gamma \) and \( \gamma' \), respectively, denote the total interval number of the disjoint union of canonical instances obtained by deleting \( E(C) \) and \( E(C') \) from \( G \) and \( G' \), respectively. To derive the claimed cost \( \sigma \), we want to prove \( I(G) - \gamma \leq I(G') - \gamma' + \sigma \) and \( I(G') - \gamma' \leq I(G) - \gamma - \sigma \). In order to do that, we show that, with adjustment by \( \sigma \) intervals, a representation of \( G' \) can be produced to model a \( u \)-optimal representation of \( G \), and similarly a representation of \( G \) can model a \( u \)-optimal representation of \( G' \). We do this by arranging locally to represent the relevant edges of \( C \) in a useful way. This is the part of the proof that differs from reduction to reduction, since it uses the canonical structure for the particular codes.

In doing so, we will let \( x \) and \( y \) denote the vertices on \( C \) immediately before and after those corresponding to the portion of the codelist for \( G \) being replaced. Note that \( x \) or \( y \) (not both) may be \( u \). We will use \( w \) for the new vertex on \( C' \) in \( G' \), or \( w^- \) and \( w^+ \) when two vertices are introduced. We will abuse notation by using \( J_i \) and \( J_{i+1} \) for intervals assigned to \( u_i \) and \( u_{i+1} \), but notation like \( J_w \) and \( J_{w^-} \) for intervals assigned to \( w \) or \( w^- \), etc.

**Lemma 5.9. Reduction 0 is valid: \((0)\) can be replaced by \((1,1)\) at no cost.**

**Proof.** We use the template in Remark 5.8. Begin with \( G \) and a fixed \( i \) with \( 1 \leq i \leq k \) such that \( c_i = 0 \). Since \( c_i = 0 \), the graph \( G_i \) is \( P_2 \) with \( u_i \) at its center, and \( f_i \) consists of a chain of five displayed intervals with a displayed interval for \( u_i \) at its center. Let \( x = u_{i-1} \) and \( y = u_{i+1} \); note that \( x \) or \( y \) may be \( u \).

Define \( G' \) from \( G \) by replacing \( u_i \) with two vertices \( w^- \) and \( w^+ \) to lengthen the cycle to \( C' \). The canonical graphs for \( w^- \) and \( w^+ \) are paths with two edges leaving \( C' \) at \( w^- \) and \( w^+ \). In a \( u \)-optimal representation of \( G' \), the \( C' \)-extraction uses three intervals in the representations \( f'_{w^-} \) and \( f'_{w^+} \) corresponding to \( w^- \) and \( w^+ \), and it makes a displayed endpoint for each of \( w^- \) and \( w^+ \) available for use in representing edges of \( C' \).
There are various ways to use these displayed endpoints to represent the edges $xw^-, w^-w^+, \text{ and } w^+y$ of $C'$. For each of these choices in the $u$-optimal $f'$, we show $I(G) - \gamma \leq I(G') - \gamma'$.

First suppose that an optimal representation $f'$ of $G'$ represents $w^-w^+$ by a hook. The hook can be formed using the end-displayed intervals in $f'_{w-}$ and $f'_{w+}$. In representing $G$, we can model this by keeping the displayed interval for $u_i$ in $f_i$ and representing $xu_i$ and $u_iy$ in whatever way $xw^-$ and $w^+y$ are represented in $f'$. In this case $I(G) - \gamma \leq I(G') - \gamma'$.

If $w^-w^+$ is not represented by a hook, then in $f'$ one of these vertices is assigned a small interval inside a displayed interval for the other. Meanwhile, the end-displayed intervals from $f'_{w-}$ and $f'_{w+}$ are available to use in representing $xw^-$ and $w^+y$. The small interval used in the insertion for $w^-w^+$ is not counted when summing the intervals used in the $C'$-extraction. In representing $G$, we can model this by breaking the displayed interval for $u_i$ in $f_i$ (recall that $G_i = P_5$) and using the resulting two end-displayed intervals in the same way that the intervals for $w^-$ and $w^+$ were used in $f'$. The interval added to form the insertion for $w^-w^+$ corresponds to the increase of one interval in breaking the interval for $u_i$; each adds one interval not counted in the $C$-extraction or $C'$-extraction. Hence again $I(G) - \gamma \leq I(G') - \gamma'$.

Now consider the reverse inequality. Using Lemma 5.4, we consider a $u$-optimal representation $f$ of $G$ such that the representation $f_i$ in the $C$-extraction is $u_i$-optimal. With $G_i = P_3$, this means that $f$ has one (displayed) interval for $u_i$ or two intervals for $u_i$. In representing $G'$, we will model each case to obtain $I(G') - \gamma' \leq I(G) - \gamma$. If $f$ uses one displayed interval for $u_i$, then represent $w^-w^+$ by a hook formed using the intervals for $w^-$ and $w^+$ in $f'_{w-}$ and $f'_{w+}$. If $f$ uses two intervals for $u_i$ (breaking the interval for $u_i$ in $f_i$), then keep $f'_{w-}$ and $f'_{w+}$ separate, adding a small interval to represent $w^-w^+$ by insertion. In either case, the edges $xw^-$ and $w^+y$ can now be represented in the same manner as $xu_i$ and $u_iy$ were represented in $f$, yielding $I(G') - \gamma' \leq I(G) - \gamma$.

In addition, besides transforming between representations of $G$ and $G'$ so that the number of intervals beyond those used in the $C$-extraction or $C'$-extraction is unchanged, we note that the $x$-extent and $y$-extent in the representation are unchanged. Hence we are obtaining $u$-optimal representations, whether or not $u \in \{x, y\}$. \hfill\(\square\)

**Lemma 5.10.** Reduction 1 is valid: $(-1, -1)$ can be replaced by (2) with cost 2.

**Proof.** We use the template in Remark 5.8. Begin with $G$ and a fixed $i$ with $1 \leq i < k$ such that $c_i = c_{i+1} = -1$. The graphs $G_i$ and $G_{i+1}$ are the single vertices $u_i$ and $u_{i+1}$ with no edges to represent. Now let $x = u_{i-1}$ and $y = u_{i+2}$. Since $f_i$ and $f_{i+1}$ are $u_i$-optimal and $u_{i+1}$-optimal, they assign no intervals.

Consider the edge $e \in E(C)$ joining $u_i$ and $u_{i+1}$, represented in $f$ by the intersection of $J_i$ and $J_{i+1}$. If one of these intervals is inserted into the other, then by symmetry we may assume $J_i$ is a small interval inside $J_{i+1}$. We can extend $J_i$ to overlap the end of $J_{i+1}$ unless
other intervals already occupy those endpoints on both sides. Since the representation is irredundant, that gives degree 3 to \( u_{i+1} \), but since \( G_{i+1} \) is empty in the canonical graph, \( u_{i+1} \) has incident edges only on \( C \). Hence we may assume that \( e \) is represented in \( f \) by a hook \( H \).

At this point both \( J_i \) and \( J_{i+1} \) are end-displayed. Their displayed endpoints are available for use in representing \( xu_i \) and \( u_{i+1}y \), respectively.

Now let \( G' \) be the canonical graph for the codelist obtained from \( c \) by replacing \((-1, -1)\) for \( \{u_i, u_{i+1}\} \) with \((2)\) for a vertex \( w \). In \( G' \) the path \((x, u_i, u_{i+1}, y)\) is replaced by edges \( wx \), \( wy \), and \( ww' \), where \( w' \) is a new vertex of degree 1. The \( w \)-optimal representation for \( G'_w \) consists of a doubly-displayed interval \( J_w \) containing a small interval \( J_{w'} \). By Lemma 5.4, in a \( u \)-optimal representation \( f' \) of \( G' \) the edge \( wu' \) is represented in this way.

We still must represent edges of the cycle at \( x \) and \( y \). No matter how we represent \( xu_i \) and \( u_{i+1}y \) in \( G \) given the form we have derived for \( f \), we can represent \( wx \) and \( wy \) in \( f' \) in essentially the same way, since the component representing \( wu' \) can be used in \( f' \) in the same way as the hook \( H \), having displayed endpoints available at both ends. Similarly, given an optimal representation \( f' \) for \( G' \), we can produce a representation for \( G \) using the hook \( H \) in the way that the component containing \( J_w \) is used to represent \( xu \) and \( wy \) in \( f' \).

The extra cost in forming \( f \) from \( f_1, \ldots, f_k \) compared to forming \( f' \) from \( f'_1, \ldots, f'_{k+1} \) is 2, because we must introduce \( J_i \) and \( J_{i+1} \) to represent the edge \( u_iu_{i+1} \). Furthermore, modeling \( u \)-optimal representations of each in forming the other yields \( I(G) - \gamma \leq I(G') - \gamma' + 2 \) and \( I(G') - \gamma' \leq I(G) - \gamma - 2 \) without changing the \( x \)-extent, \( y \)-extent, or \( u \)-extent. \( \square \)

**Lemma 5.11.** Reduction 2 is valid: \((2, -1)\) or \((-1, 2)\) can be replaced by \((2)\) with cost 1.

**Proof.** We use the template in Remark 5.8. By symmetry, begin with \( G \) and a fixed \( i \) with \( 1 \leq i < k \) such that \( c_i = 2 \) and \( c_{i+1} = -1 \). We have \( G_i = P_2 \) and \( G_{i+1} = P_2 \). Also, \( f_i \) consists of a doubly-displayed interval \( J_i \) for \( u_i \) containing a small interval for the neighbor of \( u_i \) outside \( C \). The \( C \)-extraction has no interval for \( u_{i+1} \). Since \( f \) must introduce an interval \( J_{i+1} \) for \( u_{i+1} \), we can use \( J_{i+1} \) to represent \( u_iu_{i+1} \) by a hook, leaving \( J_{i+1} \) end-displayed to be available in representing \( u_{i+1}y \) along \( C \). The other endpoint of \( J_i \) is similarly available for representing \( xu_i \).

Meanwhile, \( G' \) replaces \( u_i \) and \( u_{i+1} \) by a single vertex \( w \) with neighbor \( w' \) outside \( C' \).

In \( f' \), we have \( w \) represented by a single doubly-displayed interval \( J_w \) containing a small interval for \( w' \). Here again the displayed endpoints of \( J_w \) are available for representing the edges \( wx \) and \( wy \).

In particular, a \( u \)-optimal representation of \( G \) can thus be modeled by a representation of \( G' \), and vice versa, with cost of 1 for the introduction of \( J_{i+1} \). \( \square \)

**Lemma 5.12.** Reduction 3 is valid: \((1, -1, 1)\) can be replaced by \((1, 1)\) with cost 1.

**Proof.** We use the template in Remark 5.8. Begin with \( G \) and a fixed \( i \) with \( 1 < i < k \) such that \( c_i = -1 \) and \( c_{i+1} = c_{i-1} = 1 \). We have \( G_i = P_1 \) and \( G_{i+1} = G_{i-1} = P_3 \). In the
**Proof.**

That [Lemma 5.13](#), there is no interval for \( u_i \), while each of \( u_{i+1} \) and \( u_{i-1} \) has an end-displayed interval at the end of a segment of three intervals. We still must represent the edges of the path \( \langle x, u_{i-1}, u_i, u_{i+1}, y \rangle \).

In \( G' \) we have vertices \( w^- \) and \( w^+ \) and the path \( \langle x, w^-, w^+, y \rangle \). The \( C' \)-extraction has end-displayed intervals for \( w^- \) and \( w^+ \) at the ends of components with three intervals.

Any way of using the intervals \( J_{w-} \) and \( J_{w+} \) in representing the edges \( xw^- \) and \( w^+y \) can be modeled by using \( J_{i-1} \) and \( J_{i+1} \) in forming \( f \), and vice versa. In addition, we need to introduce small intervals for \( u_i \) to represent its edges to \( u_{i-1} \) and \( u_{i+1} \), and correspondingly a small interval for \( w^- \) or \( w^+ \) to represent \( w^-w^+ \) in \( f' \).

Alternatively, we could use \( J_{i-1} \) and \( J_{i+1} \) plus a new displayed interval for \( u_i \) to represent \( u_{i-1}u_i \) and \( u_iu_{i+1} \) by hooks, and correspondingly represent \( w^-w^+ \) by hooking \( J_{w-} \) and \( J_{w+} \). Again whatever is then done to represent \( xu_{i-1} \) and \( u_{i+1}y \) in forming \( f \) can be modeled when representing \( xw^- \) and \( w^+y \) to complete \( f' \), and vice versa.

In each case, the modeling results in one more new interval introduced when forming \( f \) from the \( C \)-extraction compared to when forming \( f' \) from the \( C' \)-extraction, so \( I(G) - \gamma = I(G') - \gamma' + 1 \). Again the modeling preserves the \( u \)-extent.

**Lemma 5.13.** Reduction 4 is valid: \((1, 1, 1)\) can be replaced by \((1)\) with cost 1.

**Proof.** We use the template in Remark 5.8. Begin with \( G \) and a fixed \( i \) with \( 1 < i < k \) such that \( c_{i-1} = c_i = c_{i+1} = 1 \). We have \( G_{i-1} = G_i = G_{i+1} = P_3 \). In the \( C \)-extraction from the \( u \)-optimal representation \( f \), each of \( \{u_{i-1}, u_i, u_{i+1}\} \) has an end-displayed interval at the end of a chain of three intervals; call these \( J_{i-1}, J_i, \) and \( J_{i+1} \). We still must represent the edges of the path \( \langle x, u_{i-1}, u_i, u_{i+1}, y \rangle \).

In \( G' \) we have a vertex \( w \) with code 1 and the path \( \langle x, w, y \rangle \). The \( C' \)-extraction has one end-displayed interval \( J_w \) for \( w \) at the end of a component with three intervals.

In forming \( f \) from its \( C \)-extraction, suppose first that neither \( u_{i-1}u_i \) nor \( u_iu_{i+1} \) is represented by a hook. Representing both by insertions requires adding two intervals not in the \( C \)-extraction. Using a hook for one of these edges saves an interval unless \( xu_{i-1} \) and \( u_{i+1}y \) are represented by hooks using \( J_{i-1} \) and \( J_{i+1} \). In that case, we can switch the representation so that \( xu_{i-1} \) is represented by an insertion and \( u_{i-1}u_i \) is represented by a hook, using the same number of intervals and not changing the \( u \)-extent.

Hence we may assume that \( u_{i-1}u_i \) or \( u_iu_{i+1} \) is represented by a hook; by symmetry suppose it is \( u_iu_{i+1} \). We can then represent \( u_{i-1}u_i \) by inserting a small interval for \( u_i \) into the displayed portion for \( u_{i-1} \), leaving that end-displayed interval available to represent \( xu_{i-1} \). This corresponds, in forming \( f' \) from its \( C' \)-extraction, to making the end-displayed interval \( J_w \) available to represent \( xw \). Representation of \( wy \) can then model the representation of \( u_{i+1}y \), and vice versa. The cost is the one small interval inserted to represent \( u_{i-1}u_i \).

**Lemma 5.14.** Reduction 5 is valid: \((2, 1, 1, 2)\) can be replaced by \((2, 2, 2)\) with cost 1.

22
Proof. Consider $G$ and a fixed segment $(2,1,1,2)$ in its codelist corresponding to vertices $v, z, z', v'$ along $C$. Let $x$ and $y$ be the other neighbors on $C$ of $v$ and $v'$, respectively. Let $\alpha, \beta, \gamma$ be the vertices replacing $\{v, z, z', v'\}$ to form $C'$ and the graph $G'$, with edges $xx\alpha$ and $\gamma y$ on $C'$. By Lemmas 5.4 and 5.6, we may let $G$ and $G'$ be the canonical graphs for their codelists. In particular, the four subgraphs replaced in $G$ are $P_2, P_3, P_3, P_2$, with the vertices on $C$ all being endpoints of these subgraphs. The graph for each of $\{\alpha, \beta, \gamma\}$ in the $C'$-extraction is $P_2$, with $\{\alpha', \beta', \gamma'\}$ denoting the respective vertices not on $C'$.

By Lemma 5.4, there is a $u$-optimal representation $f$ of $G$ whose $C$-extraction has $u_i$-optimal representations for $f_1, \ldots, f_k$. When assembling these to build $f$, we have end-displayed intervals for $z$ and $z'$ and doubly-displayed intervals for $v$ and $v'$. If $zz'$ is represented by a hook in $f$, then a new interval must be added for each of the edges $vz$ and $z'v'$. However, if $vz$ and $z'v'$ are represented by hooks using the already end-displayed intervals from the $C$-extraction, then only one small interval for $z$ or $z'$ needs to be added to represent $zz'$ by insertion. This does not affect the construction of $f$ outside these edges. Hence an optimal $f$ must not represent $zz'$ by a hook, and it must add an interval for one of $z$ or $z'$ that is not used in the $C$-extraction.

With $f$ restricted in this way, the intervals $J_v$ and $J_{v'}$ each still have a displayed endpoint available for representing the edges $xv$ and $v'y$, respectively. In $f'$, we similarly generate displayed endpoints for $\alpha$ and $\gamma$ via a chain of displayed intervals for $\alpha$, $\beta$, and $\gamma$, which contain the corresponding small intervals for $\alpha'$, $\beta'$, and $\gamma'$. Hence the completion of $f$ can be modeled by a completion of $f'$ and conversely, yielding $I(G) - \gamma \leq I(G') - \gamma' + 1$ and $I(G') - \gamma' \leq I(G) - \gamma - 1$. Again the $x$-extent, $y$-extent, and $u$-extent are unaffected. 

Lemma 5.15. Reduction 6 is valid: $(2,1,2)$ can be replaced by $(2,2)$ with cost 1.

Proof. We use the template in Remark 5.8. Begin with $G$ and a fixed $i$ with $1 < i < k$ such that $c_i = 1$ and $c_{i-1} = c_{i+1} = 2$. We have $G_i = P_3$ and $G_{i-1} = G_{i+1} = P_2$. In the $C$-extraction from the $u$-optimal representation $f$, each of $u_{i-1}$ and $u_{i+1}$ is doubly-displayed with a small interval inserted, while $u_i$ is end-displayed. The interval for $u_i$ can hook with that for $u_{i-1}$ or $u_{i+1}$, but then a small interval must be added for the edge not represented by a hook. We still have another displayed endpoint available for each of $u_{i-1}$ and $u_{i+1}$ to use in representing $xu_{i-1}$ and $u_{i+1}y$.

In $G'$, we have instead the path $(x, \alpha, \beta, y)$. In the $C'$-extraction from a $u$-optimal $f'$, we have $G_\alpha = G_\beta = P_2$, represented by insertions of small intervals for $\alpha'$ and $\beta'$ into a doubly-displayed intervals for $\alpha$ and $\beta$, respectively. Except for the cost of one small interval in $f$, we can build $f'$ by representing $x\alpha$ and $\beta y$ in the same way that $xu_{i-1}$ and $u_{i+1}y$ are represented in a $u$-optimal $f$, and vice versa.

Lemma 5.16. Reduction 7 is valid: $(2,2)$ can be replaced by $(2)$ with cost 0.
Proof. We use the template in Remark 5.8. Here $c_i = c_{i+1} = 2$, with $G_i = G_{i+1} = P_2$. In forming $f$ from the $C$-extraction, we hook $J_i$ and $J_{i+1}$ and keep displayed endpoints of both to use for $xu_i$ and $u_{i+1}y$. Similarly, the $C'$-extraction from $f'$ has $J_w$ doubly displayed to use for $xw$ and $wy$. Hence the modeling goes both directions as usual, with no added cost.

When no reductions as listed in Definition 5.7 are available, we say that the codelist is irreducible. We will need to determine the cost of each irreducible list, which as usual means the number of intervals needed to represent it beyond the number used by the $C$-extraction. We will also need the $u$-extent of a $u$-optimal representation.

Before analyzing the irreducible codelists, we describe them. Irreducible codelists have no 0, by reduction 0. Every 3-tuple from $\{-1, 1, 2\}$ is irreducible. Using symmetry, there are six distinguishable ways to choose the first and last entries, and the middle entry is arbitrary. Hence we will need to consider 18 irreducible codelists of length 3. We will show that all codelists with length at least 8 are reducible, and indeed this is also true for most lists of lengths 4 through 7.

Lemma 5.17. Up to left-right symmetry, the irreducible codelists with length at least 4 are as listed below.

<table>
<thead>
<tr>
<th>length 4</th>
<th>length 5</th>
<th>length 6</th>
<th>length 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>$[-1, 1, 1, -1]$</td>
<td>$[1, 1, 2, 1, 1]$</td>
<td>$[-1, 1, 2, 1, 1, -1]$</td>
<td>$[-1, 1, 1, 2, 1, 1, -1]$</td>
</tr>
<tr>
<td>$[1, 1, 1, 1]$</td>
<td>$[1, 1, 2, 1, -1]$</td>
<td>$[1, 2, 1, 1, -1]$</td>
<td></td>
</tr>
<tr>
<td>$[2, 1, 1, -1]$</td>
<td>$[1, 2, 1, 1, -1]$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$[1, 1, 1, 1]$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$[1, 2, 1, 1]$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$[2, 1, 1, 1]$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$[1, 2, 1, -1]$</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Proof. Consider codelists of length $k$, for $k \geq 4$. An irreducible codelist contains no 0. Also, if any two consecutive elements are in $\{-1, 2\}$, then reduction 1, 2, or 7 applies. Hence at least $[k/2]$ entries must be 1.

For $k = 4$, every codelist with at least three 1s is irreducible except $[1, -1, 1, 1]$, to which reduction 3 applies. For a codelist with two 1s, avoiding having two consecutive elements in $\{-1, 2\}$ requires the two 1s to be either (a) together in the middle, which yields an irreducible list except for $[2, 1, 1, 2]$, or (b) in positions 1 and 3 (or 2 and 4), which is subject to reduction 3 if $-1$ is between them. Hence option (b) leaves only $[1, 2, 1, -1]$ (up to symmetry), since $[1, 2, 1, 2]$ is subject to reduction 6.

For $k \geq 5$, an irreducible list cannot have three consecutive 1s, by reduction 4. Thus for $k = 5$ the only irreducible list with at least four 1s is $[1, 1, 2, 1, 1]$, avoiding reductions 3 and 4. With two 1s, they must be in positions 2 and 4, and they cannot have $-1$ between them, which then forces $-1$ on the ends to avoid reduction 6. That is, with two 1s the only irreducible list is $[-1, 1, 2, 1, -1]$. An irreducible list with three 1s cannot have them
consecutive or skipping two positions, so by symmetry their positions must be \{1, 2, 4\} or \{1, 3, 4\} or \{1, 3, 5\}. The case \{1, 3, 5\} forces 2 into positions 2 and 4 to avoid reduction 3, but then reduction 6 applies. In the other two cases, the entry between 1s must be 2 to avoid reduction 3, and then the entry in position 5 must be −1 to avoid reduction 6 or 5.

For \(k \geq 5\), every irreducible list of length \(k\) must yield an irreducible list of length \(k - 1\) when the first or last element is deleted. For \(k = 6\), consider extending an irreducible list of length 5. To avoid reductions 4 and 5, the list \([1, 1, 2, 1, 1]\) can only be extended by −1. Any extension at the end of list ending \((1, −1)\) is subject to reduction 1, 2, or 3. Hence we cannot extend \([-1, 1, 2, 1, −1]\) in either direction, and we can only extend \([1, 1, 2, 1, −1]\) or \([1, 2, 1, 1, −1]\) at the beginning. To avoid reductions 5 and 6, we cannot prepend a 2 to either. We also cannot have three consecutive 1s, by reduction 4. We can prepend −1 to \([1, 1, 2, 1, −1]\), and we can prepend −1 or 1 to \([1, 2, 1, 1, −1]\). This gives us the two lists \([1, 1, 2, 1, 1, −1]\) and \([-1, 1, 2, 1, 1, −1]\), up to symmetry.

For \(k = 7\), we have only two lists of length 6 to consider extending. All extensions admit reductions except that by prepending −1 to the list starting with 1, yielding the irreducible list \([-1, 1, 1, 2, 1, 1, −1]\). This list cannot be further extended in either direction without being subject to a reduction, so there are no irreducible lists with length at least 8.

The finiteness of the list in Lemma 5.17 implies already that a linear-time algorithm exists to compute \(u\)-optimal representations of semi-cacti. In order to make it more specific, we will analyze the irreducible lists.

**Lemma 5.18.** Let \(c_1, \ldots, c_k\) be an irreducible codelist, with \(N_i\) entries equal to \(i\), for \(i \in \{-1, 0, 1, 2\}\). Let \(G\) be the resulting graph with \(G_1, \ldots, G_k\) being the canonical representatives of graphs with codes \(c_1, \ldots, c_k\), respectively. The number \(\rho\) of intervals added to the \(C\) -extraction to obtain a \(u\)-optimal representation \(f\) of \(G\) is \(N_{−1} + 1 + \mu\), where \(\mu\) is the number of components in \(f\). Furthermore, the number of components in \(f\) is at least \(\lceil N_1 / 2 \rceil\) (and at least 1 when \(N_1 = 0\)).

**Proof.** Lemma 5.6 allows us to restrict our attention to these instances in which \(G_1, \ldots, G_k\) are the canonical representatives with codes \(c_1, \ldots, c_k\). The resulting graph \(G\) is triangle-free; in fact, \(C\) is the only cycle.

By Lemma 5.2, there is an irredundant \(u\)-optimal representation of \(G\). In exploring such a representation from left to right, each interval encountered represents a new edge, except for those that start a new component. Hence the number of intervals in the representation is the number of edges of \(G\) plus the number of components, \(\mu\). To minimize the number of intervals, we seek an irredundant representation that minimizes the number of components.

Let \(\tau\) be the total number of intervals used in the \(u_i\)-optimal representations in the \(C\)-extraction, so \(\rho = I(G) − \tau\). For the irreducible codelists, \(N_0 = 0\). Since \(G_1\) is \(P_1\), \(P_3\), or \(P_2\) when \(c_i\) is −1, 1, or 2, respectively, \(\tau = 3N_1 + 2N_2\). Since also \(|E(G)| = 2N_1 + N_2 + k + 1\)
and \( k = N_{-1} + N_1 + N_2 \), we have \( \rho = 2N_1 + N_2 + k + 1 + \mu - 3N_1 - 2N_2 \), which simplifies to \( \rho = N_{-1} + 1 + \mu \).

The term \( N_{-1} \) is natural. The C-extraction has no interval for \( u_i \) when \( c_i = -1 \), but \( u_i \) has incident edges on \( C \), so an interval for \( u_i \) must be added. If we can represent the edges incident to \( u_i \) by hooks at both ends of that interval, and similarly for the doubly-displayed interval in the C-extraction for \( u_i \) when \( c_i = 2 \), then we will not need to generate additional components.

However, if \( c_i = 1 \), then by Lemma 5.4 the component for \( G_i \) in the C-extraction consists of three intervals, with \( u_i \) singly displayed on one end. Neither of the other two intervals has a neighbor outside \( G_i \), so this forces an endpoint of a component of \( f \). With at least \( N_1 \) ends of components, we have at least \( \lceil N_1/2 \rceil \) components.

There are several advantages to having reduced the problem to irreducible codelists with canonical lobes \( G_i \). One is that we do not need to worry about the structure of \( G_i \). Another is that there are not so many indices \( i \) with \( c_i = 1 \) to consider whether the \( u_i \)-point in the representation of \( G_i \) faces left or right.

**Theorem 5.19.** Let \( G \) be the canonical graph for an irreducible codelist \( c_1, \ldots, c_k \). With \( N_i \), \( \rho \), and \( \mu \) defined as in Lemma 5.18, we have \( \rho = N_{-1} + 1 + \mu \) with \( \mu = \max\{1, \lceil N_1/2 \rceil \} \), except that \( \mu = 2 \) for the lists \([1, 2, 1, -1]\) and \([-1, 1, 2, 1, -1]\), both having \( c(u, G) = 2 \). Other special cases for the \( u \)-extent include \( c(u, G) = 0 \) for codelist \([2, 1, 2]\) and \( c(u, G) = -1 \) for codelist \([1, 2, 1]\). Outside of these exceptions, \( c(u, G) = 2 \) when \( N_1 = 0 \), \( c(u, G) = 1 \) when \( N_1 \) is odd, and \( c(u, G) = 0 \) when \( N_1 \) is positive and even.

**Proof.** The cases are as listed in Lemma 5.17, plus those for \( k = 3 \). For the nonexceptional cases, the lower bound is from Lemma 5.18, this will not be achievable in the exceptions. Let \( f_j \) be the representation for \( G_i \) in the C-extraction. When we “pass through” an interval for \( u_i \) in a component, when \( c_i = 2 \) the interval is already present in the C-extraction, but when \( c_i = -1 \) it is an added interval.

**Case** \( k = 3 \). When \( N_1 = 0 \), since all \( c_i \in \{-1, 2\} \), we make a single component starting and ending with intervals for \( u \), passing through intervals for \( u_1, u_2, u_3 \) between them. Here \( \mu = 1 \), and \( c(u, G) = 2 \).

Next \( N_1 = 1 \) with \( c_1 = 1 \) (or symmetrically \( c_3 = 1 \)). Make one component starting with \( f_1 \) and hooking through the intervals for \( u_2 \) and \( u_3 \) to end with an end-displayed interval for \( u \). The edge \( uu_1 \) is represented by insertion. This is an irredundant representation with one component, so it meets the bound. Since \( N_1 = 1 \), we cannot have such a representation in one component with \( u \) at both ends, so \( c(u, G) = 1 \).

Next \( N_1 = 1 \) with \( c_2 = 1 \). If \( c_1 = -1 \) (or symmetrically \( c_3 = -1 \)), then make one component starting with \( f_2 \) and passing through \( u_3 \) to \( u \). The edges \( uu_1 \) and \( u_1u_2 \) are
represented by insertion. Again this is an irredundant representation with one component, meeting the bound, and \( c_2 = 1 \) prevents having \( u \) at both ends, so \( c(u, G) = 1 \).

The last case with \( N_1 = 1 \) is codelist \([2, 1, 2]\). This case is an exception to the general formula for \( c(u, G) \). We obtain a representation with one component, but only by not having a \( u \)-point. If there is an irredundant representation with one component, then \( f_2 \) must hook to an interval for \( u_1 \) or \( u_3 \); by symmetry suppose it is \( u_3 \). Now \( u_2 \) does not hook to \( u_1 \) (except by introducing an extra interval for \( u_2 \). In particular, the interval at the other end of the component of \( f \) containing \( f_2 \) is not for \( u \). Therefore \( c(u, G) \leq 0 \) in an optimal representation with one component. Indeed, we can start with \( f_2 \) and pass though intervals for \( u_3 \) and \( u \) to end at \( u_1 \), representing \( uu_2 \) by insertion. Thus \( \mu = 1 \) as claimed, and \( c(u, G) = 0 \).

Now suppose \( N_1 = 2 \). If there is a representation with one component, then the vertices with code 1 provide both endpoints, forbidding a \( u \)-point. When the two entries with code 1 are consecutive, we may assume they are \( c_1 \) and \( c_2 \). We let \( f_1 \) and \( f_2 \) be the ends of the component, passing through \( u \) and \( u_3 \) between them. We can also make one component for \([1, -1, 1]\), passing through \( u \) between \( f_1 \) and \( f_2 \) and representing the edges at \( u_2 \) by inserting intervals for \( u_2 \) into the displayed intervals for \( u_1 \) and \( u_3 \). (The number of intervals added is as desired, since the representation is irredundant and has one component.) Both cases us a displayed interval for \( u \) with one component, so \( c(u, G) = 0 \).

The last case with \( N_1 = 2 \) is codelist \([1, 2, 1]\). A component with endpoints in \( f_1 \) and \( f_3 \) must pass through \( u_2 \) or \( u \). If it passes through \( u \), then the existing interval for \( u_2 \) in the \( C \)-extraction will lie in another component. Hence the component must pass through \( u_2 \), and completing a representation with only one component requires inserting small intervals for \( u \) into the displayed intervals for \( u_1 \) and \( u_3 \). Thus \( \mu = 1 \) as claimed, and \( c(u, G) = -1 \).

Finally, suppose the codelist is \([1, 1, 1]\). With three endpoints, we need two components, and achieving two components requires a fourth endpoint, which will make the representation \( u \)-optimal with \( c(u, G) = 1 \) if it is a \( u \)-point. Indeed, we can hook \( f_1 \) to \( f_2 \) and \( f_3 \) to \( u \), representing \( uu_1 \) and \( uu_3 \) by insertion.

**Case** \( k = 4 \). First consider \([1, 1, 1, 1]\). Forcing four endpoints of components requires two components, and achieving two components forbids a \( u \)-point. Hence it is optimal to hook \( f_1 \) and \( f_4 \) to \( u \) and hook \( f_2 \) and \( f_3 \) to each other, representing \( uu_2 \) and \( uu_4 \) by insertion. Here \( \mu \) is as claimed, and \( c(u, G) = 0 \).

Two cases with \( N_1 = 2 \) have \( c_2 = c_3 = 1 \), with one of the outer values being \(-1\) and the other being \(-1\) or \(2\). A representation with one component must have \( f_2 \) and \( f_3 \) at the ends, forbidding \( u \)-points and pass through \( u_1, u, u_4 \), representing \( uu_3 \) by insertion. Hence \( \mu \) is as claimed and \( c(u, G) = 0 \).

The remaining case with \( N_1 = 2 \) is \([1, 2, 1, -1]\). This is an exception to the formula for \( \mu \); we cannot produce an irredundant representation with one component. As with \([1, 2, 1]\), the component \( f_2 \) in the \( C \)-extraction would require the one component with ends in \( f_1 \) and \( f_3 \) to
pass through $u_2$, but then the edge $uu_4$ cannot be represented in that component. Hence two components are needed, and once two components are allowed the second can be a doubly-displayed interval for $u$ with small intervals for $u_1$ and $u_4$ inserted into it (another small interval for $u_4$ is inserted into the displayed interval for $u_3$). Thus $\mu = 2$ and $c(u, G) = 2$.

We are left with $N_1 = 3$, with $c_4 \in \{-1, 2\}$ or $c_2 = 2$. Two components are forced by $N_1 = 3$, with the fourth endpoint allowed to be a $u$-point. Viewing the forced endpoints and $u$ cyclically, one component hooks two of them by passing through the vertex whose code is not 1, and the other component hooks the two remaining endpoints. The edges linking the two components are represented by insertion. Thus $\mu = 2$, as claimed, and $c(u, G) = 1$.

Case $k = 5$. With $N_1 = 4$, the only case is $[1, 1, 2, 1, 1]$. Two components are forced, and achieving two components forbids a $u$-point. Hooking $f_2$ and $f_4$ to $f_3$, and hooking $f_1$ and $f_5$ to an interval for $u$ achieves two components with $c(u, G) = 0$ (two other edges are represented by insertion).

With $N_1 = 2$, the only case is $[-1, 1, 2, 1, -1]$. As in the argument for $[1, 2, 1, -1]$, we cannot represent this using one component. With two components, again we can have two $u$-points. Hook $f_2$ and $f_4$ to $f_3$, doubly-display $u$ with $u_1$ and $u_5$ inserted into it, and represent $u_1u_2$ and $u_4u_5$ by insertion. We have $\mu = 2$ and $c(u, G) = 2$.

With $N_1 = 3$, the two cases are $[1, 1, 2, 1, -1]$ and $[1, 2, 1, 1, -1]$. Two components are forced, and the fourth end among the two components allows a $u$-point. Hook the vertex with code 2 to its desired neighbors, and let the other component have endpoints at $u$ and the remaining vertex with code 1. In the latter case that component passes through $u_5$, but otherwise the remaining edges are represented by insertions. We have $\mu = 2$ and $c(u, G) = 1$.

Case $k = 6$. One case is $[1, 1, 2, 1, 1, -1]$, requiring two components. Achieving two components forbids a $u$-point. Hence connecting $u_2$ and $u_4$ through $u_3$ and connecting $u_1$ and $u_5$ through $u$ and $u_6$ produces a $u$-optimal representation with $c(u, G) = 0$.

The other case is $[-1, 1, 2, 1, 1, -1]$, also requiring two components, but allowing a $u$-point. One component connects $u_2$ and $u_4$ through $u_3$, and the other connects $u$ and $u_5$ through $u_6$, with $u_1$ inserted into both $u$ and $u_2$ (and $u_4u_5$ represented by insertion). Again $\mu = 2$, but now $c(u, G) = 1$.

Case $k = 7$. The only case is $[-1, 1, 1, 2, 1, 1, -1]$. Two components are forced and forbid a $u$-point. One connects $u_3$ and $u_5$ through $u_4$, and the other connects $u_2$ and $u_6$ through $u_1, u, u_7$, with $u_2u_3$ and $u_5u_6$ represented by insertion. We have $\mu = 2$ and $c(u, G) = 0$.

6 The Algorithm

We present a recursive algorithm to obtain a $u$-optimal representation of a semi-cactus $G$. Consider three cases:

(1) $u$ is a cut-vertex.
(2) $u$ is a non-cut-vertex in a block that is a complete graph.
(3) $u$ is a non-cut-vertex in a block that is a cycle of length at least 4.

In Case 1, with $G - u$ having $k$ components, form the $u$-lobes $G_1, \ldots, G_k$, each containing $u$. Apply the algorithm recursively to obtain a $u$-optimal representation $f_i$ of $G_i$ with code $\epsilon_u(f_i)$, for $1 \leq i \leq k$. Combine $f_1, \ldots, f_k$ according to Lemma 3.4 to obtain the representation $f$ with code $\epsilon_u(f)$. By Theorem 3.7, $f$ is a $u$-optimal representation of $G$, and $c(u, G) = \epsilon_u(f)$.

In Case 2, with $Q$ being the complete block containing $u$, suppose first that $G = Q$. In this case $I(G) = |V(Q)|$ and $c(u, G) = 2$, represented by a single pile with a doubly-displayed interval for $u$. Otherwise, let $G_1, \ldots, G_k$ be the components of $G - u - E(Q)$, with $u_i$ being the vertex of $G_i$ in $Q$. Apply the algorithm recursively to obtain for $1 \leq i \leq k$ a $u_i$-optimal representation $f_i$ of $G_i$ with code $c_i = c(u_i, G_i) = \epsilon_{u_i}(f_i)$. With $U = V(Q)$, combine $f_1, \ldots, f_k$ via $U$-completion as described in Definition 4.1 to obtain a representation $f$ of $G$, with $\epsilon_u(f)$ computed from the recursively obtained codes as in Lemma 4.2. By Theorem 4.5, $f$ is a $u$-optimal representation of $G$, and $c(u, G) = \epsilon_u(f)$.

In Case 3, with $C$ being the cycle block containing $u$, let $u_1, \ldots, u_k$ being the vertices of $C - u$ in order. Suppose first that $C = G$. In this case $I(G) = k + 2$ and $c(u, G) = 2$, represented by a chain of displayed intervals with displayed intervals for $u$ appearing at both ends. Otherwise, let $G_i$ be the component of $G - E(C)$ containing $u_i$. Apply the algorithm recursively to obtain for $1 \leq i \leq k$ a $u_i$-optimal representation $f_i$ of $G_i$, with $c_i = c(u_i, G_i) = \epsilon_{u_i}(f_i)$. Let $\tau = \sum_{i=1}^k |f_i|$.

If some reduction of the codelist as listed in Definition 5.7 is applicable, then apply such reductions to the codelist, recording a cost $\sigma_j$ as listed in Definition 5.7 for the $j$th reduction. Let $\sigma$ be the total of these costs $\sigma_j$ at the point when no further reductions are available.

If no reduction as listed in Definition 5.7 is applicable to the codelist $(c_1, \ldots, c_k)$, then compute the $u$-extent and the final cost $\rho$ using the analysis of small cases in Theorem 5.19. By Lemmas 5.9–5.16, the code for $u$ did not change in the reduction process. Let $I(G) = \tau + \sigma + \rho$. By Lemmas 5.9–5.16 and Theorem 5.19, the computation of $I(G)$ and the code $c(u, G)$ are correct. Furthermore, undoing the reductions yields the way to combine $f_1, \ldots, f_k$ to obtain the $u$-optimal representation of $G$.

7 Triangle-Free Cacti

Recall that a cactus is a connected graph in which every block is a cycle or an edge; equivalently, every edge is in at most one cycle. In this section we prove that, except for six small graphs, every $n$-vertex triangle-free cactus has total interval number at most $(18n - 12)/13$. In addition, for each $n$ the bound $\lfloor (18n - 12)/13 \rfloor$ is sharp. In the next section, we will extend the result to cover all cacti.
Our analysis proceeds by studying the parameter $h(G)$, defined by

$$h(G) = 18|V(G)| - 13I(G) - 12.$$  

The desired inequality $I(G) \leq (18|V(G)| - 12)/13$ is equivalent to $h(G) \geq 0$. We prove the bound for triangle-free cacti via several reductions to simpler instances. Define the cycle excess of a triangle-free cactus to be the sum, over all cycle blocks, of the length of the cycle minus 4. In considering a “smallest” counterexample, we minimize first the number of pendant edges, then the cycle excess, then the number of vertices.

Within the family of triangle-free cacti, our reductions replace a graph $G$ with a graph $G'$ that is earlier than $G$ in this ordering and satisfies $h(G') < h(G)$. Thus $h(G') \geq 0$ implies $h(G) \geq 0$. If all graphs are so reducible, then the claim holds inductively; that is, there is no minimal counterexample. We must be attentive, when $G$ reduces to $G'$ and $G'$ is one of the exceptional graphs, to show that $h(G) - h(G')$ is big enough to yield $h(G) \geq 0$.

In analyzing the change in $h$ via a reduction, we use the well-known expression for $I(G)$ in triangle-free graphs articulated first in [1]. For completeness, we include its proof. A trail cover of a graph $G$ is a family of pairwise edge-disjoint trails that together form a vertex cover of $G$; that is, every edge has at least one endpoint in one of the trails. The minimum number of trails in a trail cover of $G$ is the trail cover number of $G$.

Lemma 7.1 ([1]). If $G$ is a triangle-free graph with $m$ edges, then $I(G) = m + t$, where $t$ is the minimum number of trails in a trail cover of $G$.

Proof. First $I(G) \leq m + t$, since each trail can be represented using one more interval than edge (with all intervals displayed), and edges not in the trails can be represented by inserting a small interval for one endpoint into a displayed interval for the other endpoint that belongs to one of the trails.

Recall that a triangle-free graph has an optimal representation that is irredundant, meaning that each edge is represented only once by an intersection of two intervals (a second such intersection can be deleted from the representation). From an optimal irredundant representation we obtain $t \leq I(G) - m$. Since $G$ is triangle-free, no point lies in three intervals. The displayed intervals represent trails, with one trail per component. The number of small intervals is the number of edges outside the trails. Hence the resulting trails form a trail cover of size $I(G) - m$. 

Although this is a simple characterization, it is important to note that finding the trail cover number is an NP-complete problem ([7]), even on 3-regular triangle-free planar graphs (though our algorithm shows that it is easy on triangle-free cacti).

Recall that a lobe at a cut-vertex $u$ is an induced subgraph $G[S \cup \{u\}]$, where $S$ is the vertex set of a component of $G - u$. In performing a reduction, usually we will modify a lobe at a vertex $u$. We must avoid changing how the trails from the modified lobe can be
used in representing the remainder of the graph. In the language of the algorithm we have
developed, that means the code of \( u \) in the new lobe should be the same as the code of \( u \)
in the old lobe. That ensures that the contribution from the rest of the graph to \( I(G) \) and
\( I(G') \) will be the same.

Next we describe the exceptional graphs.

**Lemma 7.2.** The following six graphs violate \( h(G) \geq 0 \): \( P_2 \), \( C_4 \), and any graph consisting of
two or three blocks that are all 4-cycles. On the other hand, \( h(G) \geq 0 \) holds if \( G \) is Eulerian
and is not one of these graphs.

*Proof.* Each graph \( G \) mentioned has an Eulerian trail, so \( I(G) = |E(G)| + 1 \). The graph \( P_2 \)
has two vertices and one edge, so \( h(P_2) = 36 - 26 - 12 = -2 \).

If \( G \) is Eulerian, then every block is a cycle, and all the edges lie on a single trail. If \( G \)
has \( b \) blocks and cycle excess \( c \), then it has \( 1 + 3b + c \) vertices and \( 4b + c \) edges. Hence

\[
h(G) = 18(1 + 3b + c) - 13(1 + 4b + c) - 12 = 2b + 5c - 7.
\]

Here we have \( h(G) \geq 0 \) unless \( c = 0 \) and \( b \leq 3 \), which occurs only for the listed exceptional
graphs. It is negative for \( b \in \{1, 2, 3\} \) but always at least \(-5\).

The graphs listed above are the exceptions among triangle-free cacti. When triangles are
allowed, other exceptions arise, which are described in Section 8. In Lemmas 7.3–7.13, we
consider only triangle-free cacti.

**Lemma 7.3.** An earliest nonexceptional counterexample has no pendant edge.

*Proof.* Let \( G \) be a graph with a pendant edge at \( u \). The \( u \)-code of the lobe is 2. Form \( G' \)
from \( G \) by replacing the pendant edge with a pendant 4-cycle at \( u \), which also has \( u \)-code 2.
The replacement adds two vertices and three edges, with no change in the number of trails
needed, so \( h(G) - h(G') = 39 - 36 = 3 \). Since \( G \neq P_2 \), if \( G' \) is exceptional then \( G' \) consists
of two or three 4-cycles, in which case \( h(G) \geq 0 \) by Lemma 8.4.

Lemma 7.3 covers the case where \( G \) is a 4-cycle with a pendant edge, where \( h(G) = 0 \).

**Lemma 7.4.** In an earliest nonexceptional counterexample \( G \), any Eulerian lobe is just a
4-cycle.

*Proof.* A lobe in a cactus is an Eulerian graph if and only if all of its blocks are cycles. Let
\( H \) be an Eulerian lobe at a cut-vertex \( u \) in \( G \). If \( H \) is not a 4-cycle, then form \( G' \) from \( G \)
by replacing \( H \) with a single 4-cycle incident to \( u \). Both \( H \) and the 4-cycle are covered by
a single trail, and some trail must enter either in any trail cover. Hence \( G \) and \( G' \) have the
same trail cover number.
If $H$ has $b$ blocks and cycle excess $c$, then it has $1 + 3b + c$ vertices and $4b + c$ edges. The replacement $u$-lobe in $G'$ has four vertices and four edges. Hence

$$h(G) - h(G') = 18(3b + 1 + c - 4) - 13(4b + c - 4) = 2b + 5c - 2.$$  

Since $b + c \geq 2$, we have $h(G') < h(G)$. The decrease is 2 if $H$ consists of two 4-cycles and at least 5 if $H$ consists of one longer cycle.

If $G'$ is exceptional consisting of three cycles, then $h(G') = -1$ and $h(G) > 0$. If $G'$ consists of two cycles, then $h(G') = -3$ and $h(G) > 0$ unless $G$ consists of three 4-cycles and is also exceptional.

By Lemmas 7.3 and 7.4, we have reduced the problem to considering cacti in which every leaf block is a 4-cycle and the block neighboring any leaf block either is an edge or is a cycle that intersects at least two other blocks.

**Lemma 7.5.** An earliest nonexceptional counterexample has no leaf block at the end of a path consisting of at least two edge blocks.

**Proof.** From such a graph $G$, form $G'$ by contracting the edge of the path incident to the leaf block. In both graphs, a trail must enter the leaf block, since it is a 4-cycle, and it can extend along the path. Hence $G$ and $G'$ have the same trail cover number. Since $G$ has one more vertex and edge than $G'$, we have $h(G) - h(G') = 5$, so $h(G) \geq 0$. \hfill $\square$

An edge-block is a cut-edge; deleting it leaves two components. An edge-block is *extreme* if one of the two components obtained by deleting it has no cut-edge. If $G$ has no edge-blocks, then $G$ is Eulerian, and Lemma 8.4 applies. If $G$ has one edge-block, then both components obtained by deleting it are Eulerian, and Lemma 7.4 applies to reduce them to 4-cycles. The resulting 8-vertex graph $G$ with nine edges is covered by one trail, yielding $h(G) = 18 \cdot 8 - 13 \cdot 10 - 12 = 2$.

Hence we may assume that $G$ has at least two edge-blocks. Edge-blocks separated by maximum distance in $G$ must be extreme. By Lemma 7.4, we may assume that one component obtained by deleting an extreme edge-block is a 4-cycle. A subgraph consisting of an extreme edge-block and its pendant 4-cycle we call a *flag*. A flag is attached to the rest of the graph at its vertex of degree 1. When that vertex $u$ lies on a cycle $C$ and has degree 3, the code of $u$ relative to the flag is 1, corresponding to the trail that must originate in the 4-cycle and can extend out to $u$.

**Lemma 7.6.** An earliest nonexceptional counterexample does not have two (or more) flags incident to the same vertex.

**Proof.** Suppose that $G$ has two flags at $u$. Form $G'$ from $G$ by deleting the eight vertices in the two flags. Given any optimal representation $f'$ of $G'$, form a representation $f$ of $G$ by
adding a single component with 11 intervals to the representation corresponding to a single trail through 10 edges of the two flags. Hence $h(G) \geq h(G') + 18 \cdot 8 - 13 \cdot 11 = h(G') + 1$.

We conclude that $h(G) \geq 0$ unless $G'$ is exceptional. However, if $G'$ is exceptional, then it is Eulerian or consists of a single edge and has code 2 at any vertex. That means that the single trail passing through the two flags can be modified to touch all edges of $G$. With no extra trail needed, we save one interval, and then $h(G) = h(G') + 1$.

Lemma 7.7. An earliest nonexceptional counterexample does not have two consecutive vertices on a cycle with length at least 5 that are bivalent or attached to an Eulerian lobe.

Proof. Let $G$ be a graph with such consecutive vertices $u$ and $v$ on a cycle $C$ of length at least 5. By Lemma 7.4, an Eulerian lobe in $G$ must be a 4-cycle. Form $G'$ from $G$ by contracting $uv$ into a single vertex $z$ and appending a new 4-cycle at $z$. Since $C$ has length at least 5, $G'$ is triangle-free. Also the cycle excess of $G'$ is smaller by 1 than the cycle excess of $G$, so $G'$ is earlier than $G$.

A minimum trail cover in $G$ must reach $u$ or $v$, and hence without loss of generality it can be assumed to pass through $uv$ and be available to extend beyond $u$ and $v$ in both directions, passing also through a 4-cycle lobe attached to $u$ and/or $v$ if it is present. In $G'$, the corresponding trail can traverse the new appended 4-cycle, and $G'$ must have such a trail in a trail cover. Hence $G$ and $G'$ have the same trail cover number. Since bivalent vertices have code $-1$ and vertices in 4-cycles have code 2 in those cycles, we have three cases here, with $G$ having $n$ vertices and $m$ edges and $G'$ having $n'$ vertices and $m'$ edges.

\[
\begin{array}{cccc|c}
G & G' & n - n' & m - m' & h(G) - h(G') \\
\ldots, -1, -1, \ldots & \ldots, 2, \ldots & -2 & -3 & -36 + 39 = 3 \\
\ldots, -1, 2, \ldots & \ldots, 2, \ldots & 1 & 1 & 18 - 13 = 5 \\
\ldots, 2, 2, \ldots & \ldots, 2, \ldots & 4 & 4 & 72 - 52 = 20 \\
\end{array}
\]

Since $G'$ has at least two cycles, $h(G) \geq 0$ if $G'$ is exceptional.

Lemma 7.8. An earliest nonexceptional counterexample does not have two consecutive vertices on a cycle with length at least 5 such that one is attached to a flag and the other is attached to a flag or an Eulerian lobe.

Proof. Let $G$ be a graph with such consecutive vertices $u$ and $v$ on a cycle $C$ of length at least 5. The Eulerian lobe (if present at $u$ or $v$) is a 4-cycle, and the 4-cycle in any flag is a leaf block. Form $G'$ from $G$ by contracting $uv$ into a single vertex $w$ with neighbors $x$ and $y$ on the cycle. Since $C$ has length at least 5, $G'$ is triangle-free. The cycle excess of $G'$ is smaller by 1 than the cycle excess of $G$, so $G'$ is earlier than $G$.

The trail cover number of $G$ is larger than that of $G'$ by at most 1. If a trail in an optimal cover of $G'$ passes through $xw$ and $wy$, then it can be cut and the two ends extended separately to cover the lobes at $u$ and $v$ in $G$. Otherwise, a new trail can cover those two
lobes. (There is a possibility that \( w \) forms a trail of length 0 in the cover of \( G' \), and then no extra trail is needed, but that would only help.) With \( G \) having \( n \) vertices and \( m \) edges and \( G' \) having \( n' \) vertices and \( m' \) edges, we have \( I(G) - I(G') = m - m' + 1 \). With 1 being the code for a flag at a vertex of \( C \), we have two cases:

<table>
<thead>
<tr>
<th>( G )</th>
<th>( G' )</th>
<th>( n - n' )</th>
<th>( m - m' )</th>
<th>( h(G) - h(G') )</th>
</tr>
</thead>
<tbody>
<tr>
<td>\ldots,1,1,\ldots</td>
<td>\ldots,-1,\ldots</td>
<td>9</td>
<td>11</td>
<td>162 - 156 = 6</td>
</tr>
<tr>
<td>\ldots,1,2,\ldots</td>
<td>\ldots,-1,\ldots</td>
<td>8</td>
<td>10</td>
<td>144 - 143 = 1</td>
</tr>
</tbody>
</table>

If \( G' \) is exceptional, then it is Eulerian, and no extra trail is needed for \( G \), because the Eulerian circuit can be viewed as a single trail starting with \( wx \) and ending with \( yw \), and the two ends can be extended into the lobes at \( u \) and \( v \). This increases \( h(G) - h(G') \) by 13.

**Lemma 7.9.** An earliest nonexceptional counterexample does not have three consecutive vertices on a cycle with length at least 5 such that the outer two are attached to flags and the middle one is bivalent.

**Proof.** Let \( G \) be a graph with such consecutive vertices \( u, z, v \) on a cycle \( C \) of length at least 5. Form \( G' \) by replacing the path \( \langle u, z, v \rangle \) with the single edge \( uv \). Since \( C \) has length at least 5, \( G' \) is triangle-free. The cycle excess of \( G' \) is smaller by 1 than the cycle excess of \( G \), so \( G' \) is earlier than \( G \).

The trail cover numbers of \( G \) and \( G' \) are the same: whether the flags at \( u \) and \( v \) are covered by separate trails or a single trail in \( G' \), the same can be done in \( G \), and vice versa, since any trails emanating from the flags cover \( uz \) and \( vz \). Since \( G \) has one extra vertex and one extra edge, \( h(G) - h(G') \geq 5 \), which is sufficient even when \( G' \) is exceptional.

**Definition 7.10.** A large lobe in a cactus is a lobe containing more than one cycle. When \( G \) is a cactus, a key cycle in \( G \) is a cycle block \( C \) such that at most one component of \( G - E(C) \) is a large lobe. An ostrich is a lobe of a cactus, attached at a vertex \( w \), consisting of a key 4-cycle \( C \) through \( w \) and lobes attached to \( C \) at the vertices other than \( w \) that are successively a 4-cycle, a flag, and a 4-cycle. An ostrich has 13 vertices other than the vertex \( w \) where it attaches to the rest of the cactus.

The ostrich is an important structure in the extremal examples. To motivate the terminology, the key 4-cycle can be viewed as the body of the ostrich, with the cut-edge in the flag being the neck, the 4-cycle in the flag being the head in the sand, and the two pendant 4-cycles from the body being the wings.

**Lemma 7.11.** In an earliest nonexceptional counterexample, every key cycle \( C \) is a 4-cycle having one vertex \( w \) at which a large lobe may be attached, and at each other vertex of \( C \) the choices for lobes to attach are one 4-cycle (code 2), one flag (code 1), or nothing (code -1).
Proof. Lemmas 7.3 through 7.6 guarantee that the specified choices are the only choices for the lobes at vertices other than $w$ on a key cycle. Lemmas 7.6 through 7.9 enable us to reduce the length of a key cycle to 4. First Lemma 7.6 eliminates code 0. Next, for key cycles of length at least 5, we cannot have consecutive codes in $\{-1,2\}$, by Lemma 7.7. By Lemma 7.8 we cannot have code 1 next to code 1 or 2. The remaining possibility is alternation between codes 1 and $-1$, but that is eliminated by Lemma 7.9.

\[\square\]

**Lemma 7.12.** In an earliest nonexceptional counterexample, every key cycle with its pendant lobes forms an ostrich, attached at the bivalent vertex on its key cycle.

Proof. By Lemma 7.11, a key cycle $C$ must have length 4. We consider all possibilities for the codes at the three vertices of the key cycle other than the vertex $w$ of attachment to the rest of the graph. The possibilities for the lobes are listed in Lemma 7.11.

Let $B$ be the subgraph of $G$ consisting of the key cycle and its pendant lobes, leaving only $w$. Note that at least one lobe must be a flag, because otherwise $B$ is an Eulerian lobe from $w$, which reduces to a 4-cycle. If $B$ is only a 4-cycle, with codes $[-1,-1,-1]$, then the key cycle would be not $C$ but perhaps another block containing $w$.

We want to form $G'$ by replacing $B$ with a graph $B'$ that has the same code at $w$ in order to avoid uncertainty about the change in the trail cover number. In particular, if the code at $w$ is 1, then $B'$ is a flag. If the code at $w$ is 0, then we just delete $B$. This puts code $-1$ at $w$, but it means we are not assuming any help from the lobe.

In the first three cases, $I(G) - I(G') = m - m' + 1$, accounting for the fact that one extra trail is needed to cover $B$. In the next three cases, there is no change in the trail cover number, since the one trail covering $B$ can emerge at $w$ (code 1). Except for the last two cases, the number of trails needed to cover $B$ does not depend on the order of the codes.

The final case is the ostrich. Here only one trail is needed to cover $B$, but a single trail covering $B$ cannot have an endpoint at $w$ (it can end at any other vertex of $B$). Hence we must add 1 to the trail cover number, since $w$ may have code $-1$ or 0 in $G'$.

<table>
<thead>
<tr>
<th>codes on $C$</th>
<th>code at $w$</th>
<th>$n - n'$</th>
<th>$I(G) - I(G')$</th>
<th>$h(G) - h(G')$</th>
</tr>
</thead>
<tbody>
<tr>
<td>{1, 1, 1}</td>
<td>1</td>
<td>11</td>
<td>15</td>
<td>198 - 195 = 3</td>
</tr>
<tr>
<td>{1, 1, -1}</td>
<td>0</td>
<td>11</td>
<td>15</td>
<td>198 - 195 = 3</td>
</tr>
<tr>
<td>{1, 1, 2}</td>
<td>0</td>
<td>14</td>
<td>19</td>
<td>252 - 247 = 5</td>
</tr>
<tr>
<td>{1, -1, -1}</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>54 - 52 = 2</td>
</tr>
<tr>
<td>{1, -1, 2}</td>
<td>1</td>
<td>6</td>
<td>8</td>
<td>108 - 104 = 4</td>
</tr>
<tr>
<td>{1, 2, 2}</td>
<td>1</td>
<td>9</td>
<td>12</td>
<td>162 - 156 = 6</td>
</tr>
<tr>
<td>{2, 1, 2}</td>
<td>0</td>
<td>13</td>
<td>18</td>
<td>234 - 234 = 0</td>
</tr>
</tbody>
</table>

\[\square\]

**Lemma 7.13.** If $G$ is an $n$-vertex triangle-free cactus, then $I(G) \leq (18n - 12)/13$, except for the six graphs small graphs listed in Lemma 8.4. Furthermore, for $n \geq 3$, there is some such $G$ with $I(G) = \lfloor(18n - 12)/13\rfloor$.  

35
Proof. By Lemma 7.12, any exception to the bound not listed in Lemma 8.4 is obtained from a smaller exception by attaching an ostrich, using the special vertex \( w \) in the body of the ostrich. However, because the single trail covering an ostrich can pass through \( w \), any exception from Lemma 8.4 can be covered without using an additional trail (being Eulerian or a single edge). Hence we save a trail, and the resulting graph has already \( h(G) \geq 13 \), and no additional exception can be constructed.

We achieve \( I(G) = \lceil (18n - 12)/13 \rceil \) if and only if \( h(G) \leq 12 \). For \( 3 \leq n \leq 15 \), we present the examples below, which are not unique. Beyond \( n = 15 \), we simply add an ostrich to the example for \( n - 13 \). Let \( F_1 \) denote the flag (with five vertices). Let \( F_2 \) denote the 11-vertex graph formed from two copies of \( F_1 \) by merging their pendant vertices and then growing a path of length 2 from that vertex. Let \( F_3 \) denote the 13-vertex graph formed from three copies of \( F_1 \) by merging their pendant vertices. Let \( H \) be the graph obtained from two disjoint 4-cycles by adding one edge joining them. Let \( O \) be the ostrich. Given a graph \( G \), let \( G' \) denote a graph obtained from \( G \) by subdividing one edge.

<table>
<thead>
<tr>
<th>( n )</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>( G )</td>
<td>( P_3 )</td>
<td>( P_4 )</td>
<td>( F_1 )</td>
<td>( F_1' )</td>
<td>( F_1'' )</td>
<td>( H )</td>
<td>( H' )</td>
<td>( H'' )</td>
<td>( F_2 )</td>
<td>( F_2' )</td>
<td>( F_3 )</td>
<td>( O )</td>
<td>( O' )</td>
</tr>
<tr>
<td>( h(G) )</td>
<td>3</td>
<td>8</td>
<td>0</td>
<td>5</td>
<td>10</td>
<td>2</td>
<td>7</td>
<td>12</td>
<td>4</td>
<td>9</td>
<td>1</td>
<td>6</td>
<td>11</td>
</tr>
</tbody>
</table>

8 The Extremal Result for Cacti

When allowing triangles in cacti, we must introduce a few more small exceptions to the bound \( I(G) \leq \lceil (18n - 12)/13 \rceil \).

Definition 8.1. The dumbbell \( D \) is a cactus with three blocks consisting of a triangle and two 4-cycles, with both 4-cycles incident to the triangle. The 2-valent vertex of the triangle is the nose of the dumbbell.

Lemma 8.2. The dumbbell \( D \) has nine vertices, and \( I(D) = 11 \). It satisfies \( c(D, w) = -1 \), where \( w \) is the nose of \( D \), and attaching \( D \) by the nose to any graph \( G' \) yields a graph \( G \) such that \( h(G) = h(G') = 1 \).

Proof. The dumbbell has 11 edges, so representing it without having the three edges of the triangle represented by one interval per vertex would require at least 12 intervals. Hence we represent the triangle in a pile. Representing each 4-cycle requires five intervals, and we can save one for each 4-cycle by merging an end of its trail with an interval used for the triangle. We have described the only way to represent \( D \) using 11 edges, and in it the interval for the nose is not displayed, so \( c(D, w) = -1 \).

Consider attaching \( D \) to \( G' \) at \( w \) to form \( G \). We claim \( I(G) = I(G') + I(D) \). Let \( G_1 = G' \) and \( G_2 = D \), and for \( i \in \{1, 2\} \) let \( f_i \) be a \( w \)-optimal representation of \( G_i \), with \( b_i = c(G_i, w) \). We have \( b_2 = -1 \) and hence \( b_1 + b_2 \leq 1 \). By Corollary 3.6 and Theorem 3.7 in the development
of the algorithm, every \(w\)-reduction of \(f_1\) and \(f_2\) is a \(w\)-optimal representation of \(G\) and consists of \(I(G') + I(D)\) intervals. 

In addition to attaching dumbbells to base graphs to build exceptional graphs, we will also attach 4-cycles.

**Lemma 8.3.** Let \(G\) be obtained from a graph \(G'\) by attaching a 4-cycle at a vertex \(w\) of \(G'\) (that is, the intersection of \(G'\) and the 4-cycle is the vertex \(w\)). If \(c(G', w) \geq 0\), then \(h(G) - h(G') = 2\). If \(c(G', w) = -1\), then \(h(G) - h(G') = -11\).

**Proof.** The 4-cycle has interval number 5, and \(c(C_4, w) = 2\). The vertex \(w\) is a cut-vertex in \(G\), with lobes \(G'\) and \(C_4\). By Corollary 3.6 and Theorem 3.7, \(I(G) = I(G') + I(C_4) - 1\) if \(c(G', w) \geq 0\), while \(I(G) = I(G') + I(C_4)\) if \(c(G', w) = -1\). Thus \(c(G', w) \geq 0\) yields \(h(G) - h(G') = 18 \cdot 3 - 13 \cdot 4 = 2\), while \(c(G', w) = -1\) yields \(h(G) - h(G') = 18 \cdot 3 - 13 \cdot 5 = -11\). 

We next define a set \(E\) of cacti on which \(h\) is negative. We will show that these are the only cacti for which \(h\) is negative.

**Definition 8.4.** The set \(E\) of exceptional graphs. The set \(E\) begins with \(K_2, C_4\), two 4-cycles sharing a vertex, and three graphs consisting of three blocks that are all 4-cycles. These are the exceptional triangle-free cacti of the previous section, on which \(h\) is \(-2, -5, -3,\) or \(-1\), respectively.

Using Lemmas 8.2 and 8.3, we add to \(E\) the graphs obtained by attaching \(D\) by its nose to \(K_2\) or \(C_4\), for which \(h\) is \(-1\) or \(-4\), respectively. We call the latter graph the 3-bell. Continuing to use the lemmas, we generate additional graphs in \(E\) by starting with the 3-bell and iteratively attaching a 4-cycle and at most one dumbbell (by its nose), or iteratively attaching at most three dumbbells (by their noses). There are various ways to choose the attachment points and hence various resulting graphs, but \(h\) is negative on all of these. Further attachments make \(h\) nonnegative and hence create graphs with interval number bounded by \((18n - 12)/13\), where \(n\) is the number of vertices.

Our aim is to show \(h(G) \geq 0\) for every cactus \(G\) not in \(E\). We will use induction on the number of vertices and, within a fixed number of vertices, induction on the number of edges. Small cases satisfy the claim by inspection. In light of Section 7, in the induction step we may assume that \(G\) has a triangle.

**Definition 8.5.** Given a specified triangle \(T\) in \(G\), we let \(V(T) = \{u_1, u_2, u_3\}\), let \(L_k\) be the component of \(G - E(T)\) containing \(u_k\), and let \(G_{i,j} = L_i \cup L_j \cup u_i u_j\). Thus \(G\) is the disjoint union of \(G_{i,j}\) and \(L_k\) plus the edges \(u_k u_i\) and \(u_k u_j\), where \(\{i, j, k\} = \{1, 2, 3\}\). We call \(L_1, L_2,\) and \(L_3\) the lobes of \(T\). Finally, let \(F\) denote the set of cacti \(G\) with \(|V(G)| \equiv 5 \mod 13\).
Since 18 is relatively prime to 13, the congruence class of \( h(G) \) modulo 13 is completely determined by \( |V(G)| \). With \( 18 \equiv 5 \text{ mod } 13 \), we have \( h(G) \equiv 5n + 1 \text{ mod } 13 \) when \( G \) has \( n \) vertices (that is, \( n \equiv 8(h - 1) \)). The graphs in \( E \) have \( h \)-values between \( -5 \) and \( -1 \), so their numbers of vertices lie in \( \{4, 12, 7, 2, 10\} \) modulo 13. The class of cacti \( G \) with \( h(G) = 0 \) is particularly important; the number of vertices in such graphs is congruent to 5 modulo 13, and the set of such cacti is what we have called \( F \).

In our inductive argument, we assume that the equivalence between \( E \) and \( h < 0 \) has been proved for graphs earlier than \( G \) in the ordering.

**Lemma 8.6.** If \( T \) is a triangle in a cactus \( G \) with lobes \( L_1, L_2, L_3 \) as in Definition 8.5, then
(a) If \( L_k \) is a single vertex, then \( h(G) \geq 0 \).
(b) If \( h(G_{i,j}) < 0 \), then \( h(G) \geq 0 \).
(c) If \( h(G_{i,j}), h(L_k) \geq 0 \), and \( G_{i,j} \) and \( L_k \) are not both in \( F \), then \( h(G) \geq 0 \).

**Proof.** (a) Let \( G' = G - u_iu_j \). Since the only graphs in \( E \) that have cut-edges are \( K_2 \) and the graph obtained by attaching a dumbbell to \( K_2 \), we have \( G' \notin E \). Since \( G' \) has fewer edges than \( G \), by the induction hypothesis \( h(G') > 0 \).

Consider how the edges incident to \( u_k \) appear in an optimal representation of \( G' \). There may be two \( u_k \)-intervals, separately intersecting intervals for \( u_i \) and \( u_j \), or one \( u_k \)-interval intersecting both a \( u_i \)-interval and a \( u_j \)-interval. Since \( u_k \) has no other incident edges, in either case we can add the edge \( u_iu_j \) without needing additional intervals. In the first case, delete the \( u_k \)-interval that intersects a \( u_j \)-interval and assign another copy of the other \( u_k \)-interval to \( u_j \). In the second case, extend the intervals for \( u_i \) and \( u_j \) that intersect the \( u_k \)-interval to intersect within the \( u_k \)-interval. Thus \( I(G) \leq I(G') \), which yields \( h(G) \geq h(G') \geq 0 \).

(b) Since \( G_{i,j} \) has fewer vertices than \( G \), from \( h(G) < 0 \) we conclude \( G_{i,j} \in E \). The edge \( u_iu_j \) is a cut-edge in \( G_{i,j} \). As noted above, the only graphs in \( E \) that have a cut-edge yield a trivial component when the cut-edge is deleted. Hence \( L_i \) or \( L_j \) is a single vertex, and statement (a) applies.

(c) Combine optimal representations of \( G_{i,j} \) and \( L_k \). In the representation of \( G_{i,j} \), the edge \( u_iu_j \) is established by an intersecting \( u_i \)-interval and \( u_j \)-interval. Their intersection intersects no other interval, since \( u_i \) and \( u_j \) do not lie on a triangle in \( G_{i,j} \). Add a small interval for \( u_k \) in this intersection to represent \( u_ku_i \) and \( u_ku_j \).

We have used \( I(G_{i,j}) + I(L_k) + 1 \) intervals to represent \( G \). Since \( |V(G)| = |V(G_{i,j})| + |V(L_k)| \), we have \( h(G) = h(G_{i,j}) + h(L_k) + 13 + 12 \). If \( G_{i,j} \) and \( L_k \) do not both lie in \( F \), then \( h(G_{i,j}) + h(L_k) > 0 \), and hence \( h(G) \geq 0 \). \( \square \)

**Lemma 8.7.** Let \( T \) be a triangle in a cactus \( G \), with notation as in Definition 8.5. If the lobes \( L_1, L_2, L_3 \) of \( T \) are not all in \( E \), then \( h(G) \geq 0 \).
Proof. By Lemma 8.6(b), we may assume \( h(G_{i,j}) \geq 0 \) for any two \( i \) and \( j \) in \( \{1, 2, 3\} \). We consider cases by how many of \( \{L_1, L_2, L_3\} \) lie in \( \mathbb{E} \).

If each \( L_k \notin \mathbb{E} \), then \( h(L_k) \geq 0 \) for all \( k \), by the induction hypothesis. By Lemma 8.6(c), we have \( h(G) \geq 0 \) unless all \( L_k \in \mathbb{F} \). With \( L_i, L_j \in \mathbb{F} \), we have \( |V(G_{i,j})| \equiv 10 \mod 13 \), so \( G_{i,j} \notin \mathbb{F} \) and again Lemma 8.6(c) applies.

Now suppose \( L_k \in \mathbb{E} \) but \( L_i, L_j \notin \mathbb{E} \). As above, \( h(L_i), h(L_j) \geq 0 \), so we have \( h(G) \geq 0 \) unless \( L_i, L_j \in \mathbb{F} \), by Lemma 8.6(c). As we have observed, since \(-5 \leq h(L_k) \leq -1\), we have \( |V(L_k)| \) congruent to one of \( \{4, 12, 7, 2, 10\} \) modulo 13. With \( L_i \in \mathbb{F} \), we now have \( |V(G_{i,k})| \) congruent to one of \( \{9, 4, 12, 7, 2\} \) modulo 13. In particular, \( G_{i,k} \notin \mathbb{F} \), so Lemma 8.6(c) applies using \( L_j \) and \( G_{i,k} \).

Finally, suppose \( L_k \notin \mathbb{E} \) but \( L_i, L_j \in \mathbb{E} \). The congruence classes of \( |V(L_i)| \) and \( |V(L_j)| \) modulo 13 lie in \( \{4, -1, -6, 2, -3\} \). The sums of any two of these lie in \( \{3, -2, 6, 1, -7, 1, -4, -4, -9, -1\} \).

In particular, 5 and \(-8\) do not appear, so \( G_{i,j} \notin \mathbb{F} \). Since the same arguments as before yield \( h(L_k) \geq 0 \), Lemma 8.6(c) applies using \( L_k \) and \( G_{i,j} \).

\[
\text{Theorem 8.8. For every cactus } G, \text{ either } G \in \mathbb{E} \text{ or } h(G) \geq 0. \text{ In particular, } G \in \mathbb{E} \text{ or } I(G) \leq (|V(G)| - 12)/13.
\]

Proof. By the preceding lemmas, a smallest counterexample contains a triangle, and for every triangle in such a graph \( G \), the three lobes all lie in \( \mathbb{E} \).

A triangle in \( G \) is \textit{extreme} if at least two of its lobes have no triangles. Let \( T \) be a triangle in \( G \) minimizing the total number of triangles in the two lobes with fewest triangles. The triangle \( T \) is extreme, because if one of those two lobes contains a triangle, then that triangle contradicts the choice of \( T \).

Let \( L_1 \) and \( L_2 \) be triangle-free lobes of \( T \). Each lies in \( \mathbb{E} \). By the discussion of triangle-free members of \( \mathbb{E} \) in Section 7, each of \( L_1 \) and \( L_2 \) is \( K_2 \) or is a union of at most three edge-disjoint 4-cycles. In each case, \( c(L_1, u_1) = c(L_2, u_2) = 2 \).

Let \( G^* = L_1 \cup L_2 \cup T \). (For example, \( G^* \) is a dumbbell when \( L_1 \) and \( L_2 \) are 4-cycles.) Because a \( u_i \)-optimal representation of \( L_i \) has a \( u_i \)-point, for \( i \in \{1, 2\} \), we can represent \( G^* \) from \( u_i \)-optimal representations of \( L_1 \) and \( L_2 \) by extending intervals for \( u_1 \) and \( u_2 \) to meet and adding a small interval for \( u_3 \) in their intersection. Thus \( I(G^*) \leq I(L_1) + I(L_2) + 1 \).

If both \( L_1 \) and \( L_2 \) are single edges, then \( I(G^*) \leq 5 \), and \( G \) has four more vertices than \( L_3 \). Thus \( h(G) - h(L_3) \geq 18 \cdot 4 - 13 \cdot 5 = 7 \). Since \( h(L_3) \geq -5 \), we have \( h(G) \geq 2 \).

If both \( L_1 \) and \( L_2 \) are Eulerian, then \( G^* \) consists of a dumbbell plus up to four attached 4-cycles. Now \( G \) arises from the graph \( L_3 \) in \( \mathbb{E} \) via attaching a dumbbell (by its nose) and then attaching some number of 4-cycles. Using the description of the construction of \( \mathbb{E} \) and the lemmas in this section, we conclude \( G \in \mathbb{E} \) or \( h(G) \geq 0 \).

39
The remaining case (by symmetry) has $L_1$ as a single edge and $L_2$ as a union of $t$ 4-cycles, where $1 \leq t \leq 4$. Here $I(G^*) \leq 2 + (4t + 1) + 1 = 4t + 4$ and $G$ has $3t + 3$ more vertices than $L_3$. Thus $h(G) - h(L_3) \geq 18(3t + 3) - 13(4t + 4) = 2(t + 1)$. Since $h(L_3) \geq -5$, we have $h(G) \geq 2t - 3$. Hence $h(G)$ is negative only if $t = 1$ and $L_3$ is a 4-cycle. In this case, $G$ arises from the single edge $L_1$ via attaching (by its nose) the dumbbell consisting of $L_2 \cup L_3 \cup T$. Hence $h(G) \geq 0$ or $G \in \mathcal{E}$. 

\[\square\]

9 **The Extremal Problem for Semi-Cacti**

Semi-cacti with $n$ vertices form a larger family than cacti with $n$ vertices; it also contains the block graphs and graphs that have both cycles and cliques as blocks. We will prove $I(G) \leq [(3n - 4)/2]$ when $G$ is an $n$-vertex semi-cactus. This value is larger than for cacti. Since $[(18n - 12)/13] = [(18n - 24)/13]$, it suffices to show $(3n - 4)/2 > (18n - 24)/13$, which is valid for $n \geq 2$. In fact, the $n$-vertex semi-cacti with largest interval number are block graphs.

**Definition 9.1.** For even $n$, the sun $S_n$ with $n$ vertices is the graph formed by adding a matching of $n/2$ edges joining a complete graph with $n/2$ vertices and an independent set with $n/2$ vertices. We let \{u$_1\}, \ldots, u_{n/2}\} be the vertex set of the complete graph, \{v$_1\}, \ldots, v_{n/2}\} be the vertex set of the independent set, and \{u_i v_i: 1 \leq i \leq n/2\} be the matching. (The term “sun” has been used in various ways in the literature.)

**Lemma 9.2.** $I(S_n) = (3n - 4)/2$, where $n$ is even and $S_n$ is the $n$-vertex sun of Definition 9.1.

**Proof.** For the upper bound, we construct a representation. Make a pile of intervals for \{u$_1\}, \ldots, u_{n/2}\}, with one extending to the left and one extending to the right. Hook an interval to each of the extending intervals in order to represent the pendant edges at the two corresponding vertices. To represent the other pendant edges, add components consisting of two intersecting intervals. This uses three intervals for each pair \{u$_i, v_i\}$, except that one interval is saved for each of two pairs.

To prove $I(S_n) \geq (3n - 4)/2$, we use induction on $n$. The claim is true for $n = 2$ and $n = 4$ by inspection. The graph $S_6$ is well known to be not an interval graph; we include a proof for completeness. Consider a representation of size 6. By symmetry, we may assume that the vertices in the independent set \{v$_1, v_2, v_3$\} occur in the order $v_1, v_2, v_3$ from left to right. Since the path \(v_1, u_1, u_3, v_3\) is represented with one interval per vertex, $f(u_1) \cup f(u_3)$ contains $f(v_2)$, which introduces a forbidden edge.

For larger even $n$, consider a representation $f$ and fix a vertex pair \{u$_i, v_i\}$. Let $f'$ be the representation obtained by deleting the intervals for $u_i$ and $v_i$. Note that $f'$ is a representation of $S_{n-1}$. By the induction hypothesis, $|f'| \geq (3n - 10)/2$. If $|f| < (3n - 4)/2$,
then only one interval can be used for each of \( u_i \) and \( v_i \). The argument is valid for all \( i \), so \( S_n \) and all its induced subgraphs must be interval graphs. However, we have just shown that the subgraph \( S_6 \) induced by three of the pairs is not an interval graph.

In order to prove \( I(G) \leq (3n - 4)/2 \) when \( G \) is an \( n \)-vertex semi-cactus, we introduce a function \( h' \) that we will study as we did \( h \) in Sections 7 and 8. Let \( h'(G) = 3|V(G)| - 2I(G) - 4 \); note that \( h'(G) \geq 0 \) if and only if \( I(G) \leq (3|V(G)| - 4)/2 \).

We should be able to use the combining lemmas in the earlier sections to prove easily by induction on \( n \) that \( h'(G) \geq 0 \) when \( G \) is an \( n \)-vertex semi-cactus (for \( n \geq 2 \)). This should be easy because there are no exceptions, right?

References


