# Iterated Colorings of Graphs

Sandra M. Hedetniemi* Stephen T. Hedetniemi*
Alice A. McRae† Dee Parks† Jan Arne Telle‡

August 12, 2002

**Abstract**

For a graph property P, in particular maximal independence, minimal domination and maximal irredundance, we introduce iterated P-colorings of graphs. The six graph parameters arising from either maximizing or minimizing the number of colors used for each property, are related by an inequality chain, and in this paper we initiate the study of these parameters. We relate them to other well-studied parameters like chromatic number, give alternative characterizations, find graph classes where they differ by an arbitrary amount, investigate their monotonicity properties, and look at algorithmic issues.

## 1 Introduction

An *(undirected) graph* $G = (V, E)$ consists of a finite, nonempty set $V$ of *vertices*, and a set $E$ of unordered pairs of vertices called *edges*. Two distinct vertices $u$ and $v$ are *adjacent* if $(u, v) \in E$, and we say that $u$ is a neighbor of $v$ and $v$ is a neighbor of $u$.

Various properties can be associated with subsets of the vertices of a graph. A set $S \subseteq V$ of vertices is said to be *independent* if no two vertices in $S$ are adjacent. A set $S \subseteq V$ is called a *dominating* set if for all vertices $u \notin S$, there is a vertex $v \in S$ such that $(u, v) \in E$.

For vertex $v \in V$, the *open neighborhood* of $v$, denoted $N(v)$, is the set of vertices $u \neq v$ that are adjacent to $v$. We define the *closed neighborhood* of $v$ as $N[v] = N(v) \cup \{v\}$. The *open neighborhood of set $S$* $N(S)$ (resp. closed neighborhood $N[S]$) is the union of all the open neighborhoods $N(v)$ (resp. closed neighborhoods $N[v]$) of vertices $v \in S$. Given a set $S \subseteq V$, the *subgraph of $G$ induced by $S$* is the graph $G[S] = (S, E \cap S \times S)$.

*Department of Computer Science, Clemson University, Clemson, SC 29634
†Department of Mathematical Sciences, Appalachian State University, Boone, NC 28608
‡Department of Computer Science, University of Bergen, N-5020 Bergen, Norway

A set $S \subseteq V$ is said to be an *irredundant* set if for every vertex $u \in S$, $N[u] - N[S - \{u\}] \neq \emptyset$, that is, each vertex $v \in S$ either has no neighbor in $S$ or has at least one neighbor $w \in V - S$ that is not a neighbor of any other vertex in $S$. We refer to such a vertex $w$ as a *private neighbor* of $v$, and if $v$ has multiple private neighbors, we refer to these vertices as the *private neighbor set* of $v$.

Let $P$ be a property associated with a vertex set. We refer to a set having property $P$ as a *$P$-set*. We will assume that for all properties $P$ of interest, an isolated set of vertices $S$ has property $P$. We are often interested in finding either a maximum or a minimum cardinality $P$-set in graph $G$, or perhaps only the cardinality of a maximum or minimum $P$-set. Whether we are interested in a maximum or a minimum $P$-set depends on the property $P$. If $P$ is the property of being an independent set or an irredundant set, then the minimum $P$-set is simply the empty set $\emptyset$, so maximum $P$-sets are of interest. If $P$ is the property of being a dominating set, the maximum $P$-set is the entire set $V$, so we are interested in minimum dominating sets.

In addition to maximum and minimum $P$-sets, we can define maximal and minimal $P$-sets. A $P$-set $S$ is *maximal* if no proper superset of $S$ is a $P$-set, and is *minimal* if no proper subset of $S$ is a $P$-set.

The *vertex independence number* of $G$ is the maximum cardinality of an independent set of $G$, and is denoted $\beta_0(G)$. Because any maximal independent set is also a dominating set, we refer to the minimum cardinality of a maximal independent set as the *independent domination number* of $G$, denoted $i(G)$.

The *domination number* $\gamma(G)$ of $G$ is the minimum cardinality of a dominating set of $G$. The maximum cardinality of a minimal dominating set of $G$ is called the *upper domination number* of $G$ and is denoted $\Gamma(G)$.

Finally, the *irredundance number* $ir(G)$ is the minimum cardinality of a maximal irredundant set of $G$, and the *upper irredundance number* $IR(G)$ of $G$ is the maximum cardinality of an irredundant set of $G$.

A well-known relationship between all of these parameters is given in the following theorem by Cockayne, Hedetniemi and Miller in 1978 [5].

**Theorem 1** *For any graph $G$, $ir(G) \leq \gamma(G) \leq i(G) \leq \beta_0(G) \leq \Gamma(G) \leq IR(G)$.*

For a complete discussion of this inequality chain, the reader is referred to the book by Haynes, Hedetniemi and Slater on domination in graphs [9].

A *$k$-coloring* of a graph $G$ is simply an assignment $f : V \rightarrow \{1, 2, \ldots, k\}$ of $k$ *colors* (i.e. the integers $1, 2, \ldots, k$) to the vertices of $G$. A coloring $f$ is called *proper* if adjacent vertices are always assigned different colors.

Equivalently, a $k$-coloring is a partition $\Pi = \{V_1, V_2, \ldots, V_k\}$ of $V(G)$ into *color classes* $V_i$, where every vertex in $V_i$ is assigned the color $i$. In a proper coloring each color class $V_i$ is an independent set. The minimum number of colors in a proper coloring of a graph $G$ is called the *chromatic number* of $G$, and is denoted $\chi(G)$. More generally, a *$P$-coloring* is a partition $\Pi =$

$\{V_1, V_2, \ldots, V_k\}$ of $V(G)$ such that for every $i$, $1 \leq i \leq k$, $V_i$ is a $P$-set. In the following we shall speak of $k$-colorings either as functions $f$ or as partitions $\Pi$, and we shall require that $V_i$ is a $P$-set in the graph remaining after removing $V_1, ..., V_{i-1}$.

# 2 Iterated Coloring Algorithm

In this paper we examine some different types of $P$-colorings that arise from the Iterated Coloring Algorithm (ICA) given below, which was also studied in [12]. Let $P$ be some property associated with a set of vertices in a graph $G = (V, E)$. Algorithm ICA repeatedly removes a set $S$ of vertices having property $P$ and assigns the same color to every vertex in $S$. Each successive set $S$ is selected with respect to the graph that remains after the vertices up to and including the most recent set $S$ have been removed. These sets form *color classes* $V_1, V_2, \ldots, V_k$, where $k$, the number of colors used, is the number of sets removed before the graph becomes empty.

**Iterated Coloring Algorithm (ICA)**

*Input:* graph $G = (V, E)$, property $P$

*Output:* $P$-coloring $\{V_1, V_2, \ldots, V_k\}$

$i = 0$;
**while** ($V$ is not empty) {
    find an arbitrary $P$-set $S$ in $G[V]$;
    $i{+}{+}$;
    $V_i = S$;
    $V = V - S$;
}
$k = i$;

Notice the inherent nondeterminism in Algorithm ICA. Since it removes an arbitrary $P$-set during each iteration of the **while**-loop, many different outcomes are possible for a given graph $G$. We will be interested in the set of all possible outcomes for a graph $G$, that is, in the set of all possible $P$-colorings that Algorithm ICA can create for a given graph $G$.

Let $P$ be the property of being a maximal independent set; we write $P =$ *maximal-independent*, for shorthand. Let $G$ be the graph shown in Figure 1. The numbers assigned to the vertices represent two possible colorings that can be created by Algorithm ICA with $G$ and $P =$ *maximal independent* as input.

An assignment of colors to the vertices of any graph $G$ by Algorithm ICA with $P =$ *maximal-independent* is called an *iterated maximal-independent coloring* of $G$, or a **independent coloring*, for short [read: 'star' independent coloring]. A *independent coloring is a partition of $V$ into *independent color*

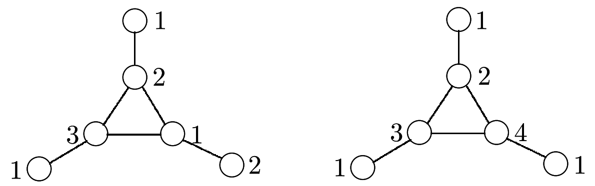



Figure 1: Colorings created with $P$ = *maximal-independent*

*classes* $\{V_1, V_2, \ldots, V_k\}$, where each $V_i$ is a maximal independent set in the graph $G_i = G - V_1 - V_2 - \cdots - V_{i-1}$. These colorings were first defined by Prins in 1963 [11], who called them *Type-1 colorings.* The minimum number of colors used over all runs of Algorithm ICA, with inputs $G$ and $P$ = *maximal independent*, is called the *iterated independent domination number* of $G$, and is denoted by $i^*(G)$. The maximum number of colors used over all runs of Algorithm ICA, with $G$ and $P$ = *maximal independent* as input, is denoted $\beta_0{}^*(G)$, and is called the *iterated independence number* of $G$.

**Theorem 2** *For any graph $G$, $i^*(G) = \chi(G)$.*

**Proof.** Because a *independent coloring is a proper coloring, it follows that $\chi(G) \leq i^*(G)$.

Conversely, it can be shown that for any proper coloring of a graph $G$ with $k$ colors, there exists a *independent coloring of $G$ with at most $k$ colors. Let $\Pi = \{V_1, V_2, \ldots, V_k\}$ be any proper coloring of a graph $G$. If $V_1$ is not a maximal independent set of $G$, then create a maximal independent set containing $V_1$ by moving vertices in $V_2$, if any, that are not adjacent to any vertices in $V_1$ into $V_1$. Call the resulting set $V_{12}$. Next, move any vertices in $V_3$ into $V_{12}$ if they are not adjacent to any vertices in $V_{12}$. Call the resulting set $V_{13}$. Repeat this process iteratively for every set $V_4$ through $V_k$. At this point the resulting set $V_{1k}$ will be a maximal independent set of $G$.

Next, let $V_2'$ be the set of vertices in $V_2$ that remain after the process of creating $V_{1k}$. Repeat the process of moving vertices from higher indexed sets into $V_2'$ if possible, resulting in a set $V_{2k}$ which is maximal independent in the graph $G_1 = G - V_{1k}$. This process can be continued for all remaining sets. The resulting coloring will then be a *independent coloring with at most $k$ colors.

It follows from this argument, that if the original proper coloring had been a coloring with $k = \chi(G)$ colors, then the resulting *independent coloring will have at most $\chi(G)$ colors. Thus, $i^*(G) \leq \chi(G)$, and hence, $i^*(G) = \chi(G)$. $\Box$

We can also equate the iterated independence number, $\beta_0^*(G)$ with a well-known coloring invariant. Let $\Pi = \{V_1, V_2, \ldots, V_k\}$ be any proper coloring of a graph, and let $v \in V_j$, for some index $j$. We say that $v$ is a *Grundy vertex* if it is adjacent to at least one vertex $u \in V_i$, for every $i$, $1 \leq i < j$. Notice that every vertex in $V_1$ is a Grundy vertex. We say that a proper coloring $\Pi$ is a Grundy coloring if every vertex is a Grundy vertex. The maximum number of colors used in a Grundy coloring of a graph $G$ is called the *Grundy coloring number* of $G$, and is denoted $GN(G)$. The Grundy number of a graph is well-studied, see [8, 4, 3, 7, 10], for example.

**Theorem 3** *For any graph $G$, $\beta_0^*(G) = GN(G)$.*

**Proof.** The proof is simple, since every Grundy coloring $\Pi = \{V_1, V_2, \ldots, V_k\}$ of a graph $G$ is an iterated maximal-independent coloring that can be created by Algorithm ICA. In particular, it can be seen that $V_i$ must be a maximal independent set in $G_i = G - V_1 - V_2 - \ldots - V_{i-1}$.

Conversely, it is easily seen that given any *independent coloring, every vertex is a Grundy vertex. □

We can create other kinds of colorings using the Algorithm ICA if we change property $P$. Let $G$ be a graph and let $P =$ *minimal-dominating*. An assignment of colors to the vertices of $G$ by Algorithm ICA is called an *iterated minimal-dominating coloring* of $G$, or a **dominating coloring*, for short. A *dominating coloring is a partition of $V$ into color classes $V_1, V_2, \ldots, V_k$, each of which is a minimal dominating set of vertices in the graph $G_i = G - V_1 - V_2 - \cdots - V_{i-1}$. The fewest number of colors used over all runs of Algorithm ICA is called the *iterated domination number* of $G$, and is denoted by $\gamma$*(G)*. The largest number of colors used is called the *iterated upper domination number* of $G$, and is denoted by $\Gamma^*(G)$.

Since every maximal independent set $S$ is a minimal dominating set, we know that

$$\gamma^*(G) \leq i^*(G) = \chi(G) \leq \beta_0^*(G) = GN(G) \leq \Gamma^*(G) \tag{1}$$

Now let $P =$ *maximal-irredundant* and $G$ be any graph. An assignment of colors to the vertices of $G$ by Algorithm ICA is called an *iterated maximal-irredundant coloring* of $G$ (or a *irredundant coloring, for short), and is a partition of $V$ into color classes $V_1, V_2, \ldots, V_k$, each of which is a maximal irredundant set in the graph $G_i = G - V_1 - V_2 - \cdots - V_{i-1}$. The largest number of colors used over all runs of Algorithm ICA is called the *iterated upper irredundance number* of $G$, and is denoted by *IR**(G)*. The minimum number of colors used is called the *iterated irredundance number* of $G$, and is denoted by *ir**(G)*.

Since every minimal dominating set $S$ is a maximal irredundant set, we have shown the following:

**Theorem 4** *For any graph $G$*

$$ir^*(G) \leq \gamma^*(G) \leq i^*(G) \leq \beta_0^*(G) \leq \Gamma^*(G) \leq IR^*(G)$$

# 3 Alternative characterizations

Two properties completely characterize iterated maximal-independent colorings. One of these properties guarantees that each of the color classes is an independent set, and the other guarantees that each independent set is maximal in the remaining graph $G_i$. Likewise there are two properties that characterize iterated minimal-dominating colorings; one guarantees that each of the color classes is a dominating set in $G_i$, and one guarantees that each dominating set is minimal. Because every maximal independent set is a minimal dominating set, the maximality property for independent colorings is very similar to the domination property for iterated minimal-dominating colorings. In similar fashion, there are two properties that characterize iterated maximal-irredundant colorings, one that guarantees irredundance and one that guarantees maximality. Since every minimal dominating set is maximal irredundant, the minimality property for iterated minimal-dominating colorings becomes the irredundance property for iterated maximal-irredundant colorings.

In order to describe the graph $G$ as Algorithm ICA removes vertices, we use the notation $G_i = (V, E_i)$ to represent the graph remaining at the start of iteration $i$. The initial graph $G$ is $G_1$. The vertices in color class $V_i = S$ are removed from $G_i$ during iteration $i$. The final coloring is denoted $\Pi = \{V_1, V_2, \ldots, V_k\}$.

## 3.1 Iterated maximal-independent colorings

In this section we describe two properties that completely characterize *independent colorings of a graph $G$. We shall show that these properties are both necessary and sufficient to characterize such colorings.

**Lemma 1** *Let $f : V \to \{1, 2, \ldots, k\}$ be a *independent coloring of $G = (V, E)$. Then for any two adjacent vertices $u$ and $v$, $f(u) \neq f(v)$.*

**Proof.** Suppose there are two adjacent vertices $u$ and $v$ with $f(u) = f(v) = i$. Then color class $V_i$ is not an independent set, which is a contradiction. □

**Lemma 2** *Let $f : V \to \{1, 2, \ldots, k\}$ be a *independent coloring of $G = (V, E)$. Then for every vertex $v \in V$ with $f(v) > 1$ and for all $i$, $1 \leq i < f(v)$, there is a vertex $w$ in the neighborhood of $v$ with $f(w) = i$, that is, $v$ is a Grundy vertex.*

**Proof.** Suppose it is not true that for every vertex $v \in V$ with $f(v) > 1$ there is a vertex $w$ in the neighborhood of $v$ with $f(w) = i$ for all $i$, $1 \leq i < f(v)$. Then there must be a vertex $u \in V$ with $f(u) > 1$ and an $i$, $1 \leq i < f(u)$, such that there is no vertex $x \in N(u)$ with $f(x) = i$. But $V_i$ is a maximal independent set in $G_i$, and as such it is also a dominating set for $G_i$. Since $u \in V[G_i]$, some vertex with color $i$ dominates $u$, i.e., some vertex with color $i$ is adjacent to $u$. This is a contradiction. □

**Theorem 5** *The two properties described in Lemmas 1 and 2 are necessary and sufficient to characterize a coloring of a graph $G$ created by Algorithm ICA when $P$ = maximal-independent.*

**Proof.** The lemmas given above show that the properties are necessary for a *independent coloring. We now show that they are sufficient. Given a coloring $\Pi$ of graph $G$ for which both properties hold, we must show that $\Pi$ could have been created by the Algorithm ICA with $P$ = *maximal-independent.* In other words, we must show that for any $i$, $V_i$ is a maximal independent set for graph $G_i$. Suppose this is not true. If $V_i$ is not an independent set, then there must exist two vertices in $V_i$ that are adjacent. But Lemma 1 says that any two adjacent vertices must have different colors, which yields a contradiction. Suppose therefore that $V_i$ is not a maximal independent set, i.e., that $V_i$ is not a dominating set. Then there exists a vertex $w$ in $G_i$ that is not adjacent to any vertex colored $i$. This vertex $w$ will be colored with a color greater than $i$. By Lemma 2, any vertex with a color greater than 1 is adjacent to vertices of all colors less than its own color, making this a contradiction. □

## 3.2 Iterated minimal-dominating colorings

In this section we describe two properties that completely characterize *dominating colorings of a graph $G$. We show that these properties are necessary and sufficient to characterize such colorings.

**Lemma 3** *Let $f : V \to \{1, 2, \dots, k\}$ be a *dominating coloring of $G = (V, E)$. Then every vertex $v \in V$ is a Grundy vertex, that is, for all $i$, $1 \leq i < f(v)$, there is a vertex $w \in N(v)$ with $f(w) = i$.*

**Proof.** Suppose it is not true that every vertex $v \in V$ is a Grundy vertex. Then there must be a vertex $u \in V$ with $f(u) > 1$ and an $i$, $1 \leq i < f(u)$, such that there is no vertex $x \in N(u)$ with $f(x) = i$. But $V_i$ is a dominating set for $G_i$, and therefore some vertex with color $i$ must dominate $u$, i.e., some vertex with color $i$ is adjacent to $u$. This is a contradiction. □

**Lemma 4** *Let $f : V \to \{1, 2, \dots, k\}$ be a *dominating coloring of $G = (V, E)$. If two adjacent vertices $v$ and $w$ are colored with the same color $i < k$, there must exist distinct vertices $y \in N(v)$ and $z \in N(w)$ such that $f(y)$ and $f(z)$ are both greater than $i$, and neither $y$ nor $z$ is adjacent to another vertex colored $i$.*

**Proof.** The set $V_i$ is a minimal dominating set of $G_i$. Since any minimal dominating set is also an irredundant set, if vertices $v$ and $w$ are adjacent, then they must each have private neighbors in $G_i$. □

**Theorem 6** *The two properties described in Lemmas 3 and 4 are necessary and sufficient to characterize the colorings of a graph $G$ created by Algorithm ICA when $P$ =* minimal-dominating.

**Proof.** Lemmas 3 and 4 show that the properties are necessary for a *dominating coloring. We now show that they are sufficient. Given a coloring $\Pi = \{V_1, V_2, \ldots, V_k\}$ of a graph $G$ for which both properties hold, we must show that $\Pi$ can be created by Algorithm ICA with $P =$ *minimal-dominating*. In other words, we must show that for any $i$, $V_i$ is a minimal dominating set for graph $G_i$.

Suppose this is not true. If $V_i$ is not a dominating set, then there must exist some vertex in $G_i$ that is not dominated by $V_i$. But every vertex in $G_i$ that is not colored $i$ must be adjacent to a vertex colored $i$ (by Lemma 3), which yields a contradiction. Suppose $V_i$ is not a minimal dominating set, i.e., that $V_i$ is not irredundant. Then there exists a vertex $w$ colored $i$ that does not have a private neighbor. Clearly $w$ is adjacent to some other vertex colored $i$, or $w$ would have itself as a private neighbor. By Lemma 4 any two adjacent vertices of the same color have private neighbors, making this a contradiction. $\square$

## 3.3 Iterated maximal-irredundant colorings

In this section we describe two properties that completely characterize *irredundant colorings of a graph $G$. We shall show that these properties are both necessary and sufficient to characterize such colorings.

**Lemma 5** *Let $f : V \to \{1, 2, \ldots, k\}$ be a *irredundant coloring of $G = (V, E)$. Then if two adjacent vertices $v$ and $w$ are colored with the same color $i < k$, there must exist distinct vertices $y \in N(v)$ and $z \in N(w)$ such that $f(y)$ and $f(z)$ are both greater than $i$, and neither $y$ nor $z$ is adjacent to another vertex colored $i$.*

**Proof.** The set $V_i$ is an irredundant set for graph $G_i$. Thus any two adjacent vertices $v, w \in V_i$ must have private neighbors, say $y$ and $z$, respectively, in $G_i$. $\square$

**Lemma 6** *Let $f : V \to \{1, 2, \ldots, k\}$ be a *irredundant coloring of $G = (V, E)$. For any vertex $p$ with $f(p) > 1$ and every color $1 \leq i < f(p)$, at least one of the following must hold:*

1. *In graph $G_i$, $N[p] - N[V_i] = \emptyset$, that is, vertex $p$ and every neighbor of $p$ colored greater than $i$ are adjacent to some vertex colored $i$.*
2. *There exists a vertex $q$ in $V_i$, such that in the graph $G_i$, $N[q] - N[V_i - \{q\}] \subseteq N[p]$. That is, there is a vertex $q$ colored $i$ whose entire private neighbor set is in $N[p]$.*

**Proof.** Consider any vertex $p$ with $f(p) > x$. Since $V_x$ is a maximal irredundant set in $G_x$, the set $V_x \cup \{p\}$ is not irredundant. This set is not irredundant either because vertex $p$ would have no private neighbor with respect to $V_x$ or because vertex $p$ would destroy the private neighbor set for some vertex $q \in V_x$. $\square$

**Theorem 7** *The two properties described in Lemmas 5 and 6 are necessary and sufficient to characterize colorings of a graph $G$ created by Algorithm ICA when $P =$* maximal-irredundant.

**Proof.** Lemmas 5 and 6 show that the properties are necessary for a *irredundant coloring. We now show that they are sufficient. Given a coloring $\Pi$ of a graph $G$ for which both properties hold, we must show that $\Pi$ can be created by Algorithm ICA with $P =$ *maximal-irredundant*. In other words, we must show that for any $i$, $V_i$ is a maximal irredundant set in graph $G_i$. Suppose this is not true. If $V_i$ is not an irredundant set, then there must exist some vertex $w \in V_i$ that does not have a private neighbor in $G_i$. Clearly, $w$ must be adjacent to some other vertex in $V_i$ or $w$ would have itself as a private neighbor. By Lemma 5, any two adjacent vertices of the same color have private neighbors, which yields a contradiction.

Suppose $V_i$ is not a maximal irredundant set in $G_i$. Then there is some vertex $w$ in $G_i - V_i$ such $V_i \cup \{w\}$ is irredundant. Vertex $w$ must have a color greater than $i$ and therefore greater than 1, so by Lemma 6 one of the following conditions must hold:

1. For every color $j$ less than the color of $w$ (this includes $i$), vertex $w$ and every neighbor of $w$ colored greater than $j$ are adjacent to some vertex colored $j$. In this case, $w$ could not be added to $V_i$, a contradiction.

2. For every color $j$ less than the color of $w$ there is a vertex $q$ colored $j$ whose entire private neighbor set is in $N[w]$. Similarly in this case, $w$ could not be added to $V_i$, another contradiction.

□

# 4 Relationships Between Colorings

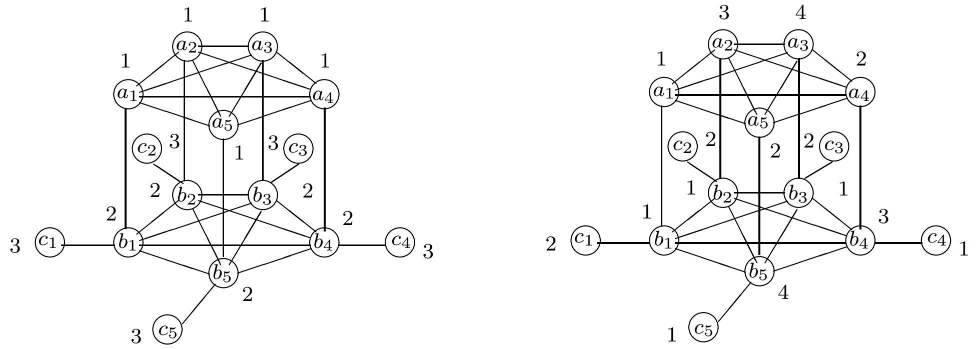
In this section we show that arbitrarily large differences can exist between each consecutive pairs of invariants in the inequality sequence:

$$ir^*(G) \leq \gamma^*(G) \leq i^*(G) \leq \beta_0^*(G) \leq \Gamma^*(G) \leq IR^*(G).$$

We use the term *endvertex* to describe a vertex having only one neighbor.

**Lemma 7** *There can be an arbitrarily large difference between $ir^*(G)$ and $\gamma^*(G)$.*

**Proof.** Consider the graph $G$ on $n = 3q$ vertices, $q \geq 3$, where $V = \{a_1, a_2, \ldots, a_q, b_1, b_2, \ldots, b_q, c_1, c_2, \ldots, c_q\}$. Form a complete graph among the $a$-vertices and another complete graph among the $b$-vertices. Add the edges $\{a_i, b_i\}$ and $\{b_i, c_i\}$, for $1 \leq i \leq q$. See Figure 2. For this graph, we will show that $ir^*(G) = 3$ and $\gamma^*(G) = \lfloor q/2 \rfloor + 2$.



$\mathrm{ir}^*(G) = 3$ $\gamma^*(G) = \lfloor 5/2 \rfloor + 2 = 4$

Figure 2: Graph where $\gamma^*(G)$ is greater than $\mathrm{ir}^*(G)$.

1. A *irredundant coloring using three colors can be found by assigning all the $a$-vertices the color 1, all the $b$-vertices color 2, and all the $c$-vertices color 3. Therefore, $ir^*(G) \leq 3$. If $ir^*(G) \leq 2$, then at least two $a$-vertices must have the same color, and they must get their private neighbors from the $b$-vertices. These $b$-vertices are adjacent and either are assigned different colors or are assigned the same color, getting their private neighbors from the $c$-vertices. In either case, at least three colors must be used.
2. The assignment $a_1 = 1$; $a_i = 2$, for $\lceil q/2 \rceil < i \leq q$; $a_i = i + 1$, for $2 \leq i \leq \lceil q/2 \rceil$; $b_i = 1$, for $1 \leq i \leq \lceil q/2 \rceil$; $b_i = i - \lceil q/2 \rceil + 2$, for $\lceil q/2 \rceil < i \leq q$; $c_i = 2$, for $1 \leq i \leq \lceil q/2 \rceil$; and $c_i = 1$, for $\lceil q/2 \rceil < i \leq q$ , is a *dominating coloring that uses $\lfloor q/2 \rfloor + 2$ colors.
3. Let $f$ be a *dominating function for $G$. To show that $\gamma(G) \geq \lfloor q/2 \rfloor + 2$, requires several observations:
   (a) Either $f(b_i) = 1$ or $f(c_i) = 1$, $1 \leq i \leq q$, otherwise vertex $c_i$ is not dominated by $V_1$. If $f(b_i) = 1$, then $f(c_i) = 2$, since vertex $c_i$ is an isolate in $G_2$.
   (b) At most one $a$-vertex can be colored 1, since no $b$-vertex can be used as a private neighbor with respect to $V_1$.
   (c) If $\gamma(G) < q$, then at least two $a$-vertices must be assigned the same color, using $b$-vertices as private neighbors. Let $x$ be the least color such that $f(a_i) = f(a_j) = x, a_i \neq a_j$. In this case, no $b$-vertices can be colored $x$. All remaining $b$-vertices must be dominated in $G_x$, so all must be adjacent to $a$-vertices that are colored $x$. Note that when $V_x$ is removed, $G_{x+1}$ will contain two disjoint cliques. Let $R$ be the number of $a$-vertices colored $x$. All $b$-vertices remaining in $G_x$ must be assigned a color greater than x, and no two vertices within a

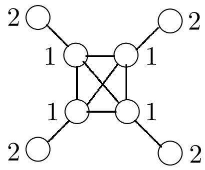



Figure 3: Graph where $\gamma^*(G) = 2$ and $i^*(G) = 4$

remaining clique can be assigned the same color. Therefore, at least one of the $b$-vertices must be assigned a color $y \geq R+2$. Considering all the $a$-vertices in $G$, R of these vertices are colored $x$, and no other vertices can have the same color. So $q - R + 1$ colors are used for the $a$-vertices. If $R < \lfloor q/2 \rfloor$, then the $a$-clique will require at least $\lfloor q/2 \rfloor + 2$ colors. If $R \geq \lfloor q/2 \rfloor$, then the $b$-clique will require at least $\lfloor q/2 \rfloor + 2$ colors. Therefore, $\gamma^*(G) = \lfloor q/2 \rfloor + 2$.

□

**Lemma 8** *There can be an arbitrarily large difference between* $\gamma^*(G)$ *and* $i^*(G)$.

**Proof.** Consider a graph $G$ on $n = 2q$ vertices that contains

- a clique of $q$ vertices, and
- $q$ endvertices, each adjacent to a distinct $q$ clique vertex.

Figure 3 shows a graph with a clique of $q = 4$ vertices. If we use Algorithm ICA with $P =$ *minimal-dominating*, all of the vertices in the clique can be colored 1, because each of them has a private neighbor. Once the vertices in the clique are removed, we color all of the remaining single vertices 2, so $\gamma^*(G) = 2$ for any such graph. However, if we use Algorithm ICA, with $P =$ *maximal-independent*, on a graph of this type, we require $q$ colors, since each of the vertices in the clique must be assigned a different color. By varying the size of the clique, we can create a graph $G$ with an arbitrarily large difference between $\gamma^*(G)$ and $i^*(G)$. □

**Lemma 9** *For any positive integer* $q$ *there is a tree* $T$ *on* $n = 2^q$ *vertices with* $i^*(T) = 2$ *and* $\beta_0^*(T) = \log n + 1$.

**Proof.** Consider the binomial tree $T = B_q$ with $n = 2^q$ vertices, which can be defined iteratively as follows:

- $B_0$ consists of one vertex
- $B_i$ consists of a copy of $B_{i-1}$ with each vertex having a new endvertex as a neighbor.

Since $B_q$ is a tree it has chromatic number $\chi(B_q) = i^*(B_q) = 2$. By the above definition it is clear that the leaves of $B_q$ form a maximal independent set and that after removing these endvertices we are left with $B_{q-1}$. Since $\beta_0^*(B_0) = 1$, we have $\beta_0^*(B_q) \geq q+1 = \log n + 1$. In fact, equality can be shown to hold. □

**Lemma 10** *For any positive integer $q$ there is a bipartite graph $G$ on $n = 2q$ vertices with $\beta_0^*(G) = 2$ and $\Gamma^*(G) = q+1$.*

**Proof.** Consider the complete bipartite graph $K_{q,q}$. It has only two maximal independent sets, namely the two vertex partition classes. After removing any of these, we are left with a graph without any edges, and thus $\beta_0^*(K_{q,q}) = 2$. On the other hand, any pair of adjacent vertices form a minimal dominating set of $K_{q,q}$. After removal of such a pair, we are left with $K_{q-1,q-1}$. Since $\Gamma^*(K_{1,1}) = 2$, we have $\Gamma^*(K_{q,q}) \geq q+1$. In fact, equality can be shown to hold. □

**Lemma 11** *For any positive integer $q$ there is a graph $G$ on $n = 2q$ vertices with $\Gamma^*(G) = 3$ and $IR^*(G) = q+1$.*

**Proof.** Consider the graph $G_q$ with two non-adjacent vertices each of degree $q-1$ and $2q-2$ vertices of degree two, obtained for example by starting with a two-vertex graph with $q-1$ multiple edges connecting these two vertices and then subdividing each edge twice. The degree two vertices induce a matching on $q-1$ edges. Any minimal dominating set of $G_q$ must for each edge $uv$ in this matching contain either $u$ or $v$ or both of the degree $q-1$ vertices. After removing such a set, the components of the remaining graph consists of star graphs $K_{1,i}$ with a center and $i$ leaves, $0 \leq i \leq q-2$. Since a minimal dominating set in a star graph consists either of the center or all of the leaves, we have $\Gamma^*(G_q) = 3$. On the other hand, any two adjacent degree-two vertices $u$ and $v$ of $G_q$ form a maximal irredundant set, since taking any additional vertex would leave either $u$ or $v$ without a private neighbor. After removal of $u$ and $v$ we are left with $G_{q-1}$. Since $G_2$ is a path on four vertices, we have $\mathrm{IR}^*(G_2) = 3$ so that $\mathrm{IR}^*(G_q) \geq q+1$, and, in fact, equality can be shown to hold. □

The following result was observed by Fricke and Hedetniemi but has never been published:

**Lemma 12** *For every tree $T$, $\beta_0^*(T) = \Gamma^*(T)$.*

**Proof.** Assume the contrary. Let $T^0$ be a smallest tree for which $\beta_0^*(T) < \Gamma^*(T)$. Let $S$ be a minimal dominating set of $T^0$ whose removal results in a forest $F$ such that $\Gamma^*(T^0) = 1 + \Gamma^*(F)$. Let $T_1$ be a tree in $F$ such that $\Gamma^*(T_1) = \Gamma^*(F)$. Since $T_1$ is smaller than $T^0$, $\beta_0^*(T_1) = \Gamma^*(T_1)$.

Let $Y$ be the set of all vertices in $S$ that are adjacent to vertices in $T_1$. Notice that $|Y| \geq |T_1|$, since every vertex of $T_1$ is adjacent to a distinct vertex in $Y$. No two vertices in $Y$ can be adjacent else $T^0$ contains a cycle. Now let $S^0$ be any maximal independent set of $T^0$ containing $Y$. Notice that $S^0 = T - T_1$ and that $T_1$ is a tree in $T - S^0$. Thus, $\beta_0^*(T) \geq 1 + \beta_0^*(T_1) = \Gamma^*(T)$; i.e., $\beta_0^*(T) = \Gamma^*(T)$. $\Box$

# 5 Monotonicity Properties of Parameters

It is well-known that removing edges from a graph cannot increase its chromatic number. We say that a graph parameter is *monotone* if it has this property: its value for a graph $H$ is at least as much as its value for any subgraph of $H$. In this section we study the monotonicity of iterated coloring parameters, and show that $i^*$ and $\Gamma^*$ are monotone, while $\mathrm{ir}^*, \gamma^*, \beta_0^*$ and $\mathrm{IR}^*$ are not monotone.

**Lemma 13** *The iterated independent domination number $i^*$ and the iterated upper domination number $\Gamma^*$ are monotone.*

**Proof.** Since $i^*$ is equal to the chromatic number it is a monotone parameter. We prove that $\Gamma^*$ is monotone.

Claim: For any graph $G = (V, E)$, supergraph $H = (V \cup W, E \cup F)$ and minimal dominating set $S$ of $G$, there exists a minimal dominating set $S'$ of $H$ such that $S' \cap V \subseteq S$.

We first show that if this claim holds, then the lemma follows. Let $\Gamma^*(G) = k$ and let $V_1, V_2, ..., V_k$ be a corresponding partition of $V$ with $V_i$ a minimal dominating set of $G_i$, the graph remaining after the removal of $V_1, V_2, ..., V_{i-1}$. The claim states that in the supergraph $H$ of $G$ we can find a minimal dominating set $V_1'$ of $H$ with $V_1' \cap V \subseteq V_1$. After the removal of $V_1'$ from $H$ we have a supergraph $H_1$ of $G_1$, and again the claim states the existence of a minimal dominating set $V_2'$ of $H_1$ with $V_2' \cap V \subseteq V_2$. Repeatedly applying this argument we can conclude that $\Gamma^*(H) \geq \Gamma^*(G)$, showing that $\Gamma^*$ is a monotone parameter.

Proof of claim: We remove vertices from $S \cup (W - V)$ to give us $S'$ by executing the following procedure, where all neighborhood references are to the supergraph $H$:

- Set $S' = S \cup (W - V)$
- **while** $\exists v \in S'$ such that $N(v) \cap S' \neq \emptyset$ and $v$ has no private neighbor in $V - S'$
  **do** remove $v$ from $S'$

The **while**-loop clearly has the invariant: *$S'$ is a dominating set of $H$*, since initially $S \subseteq S'$, $S$ is a dominating set of the subgraph $G$, and any vertex of $H$ not in $G$ is in $S'$. The invariant is maintained since whenever we remove a vertex $v$ from $S'$ it has a neighbor in $S'$ and any of its neighbors also have

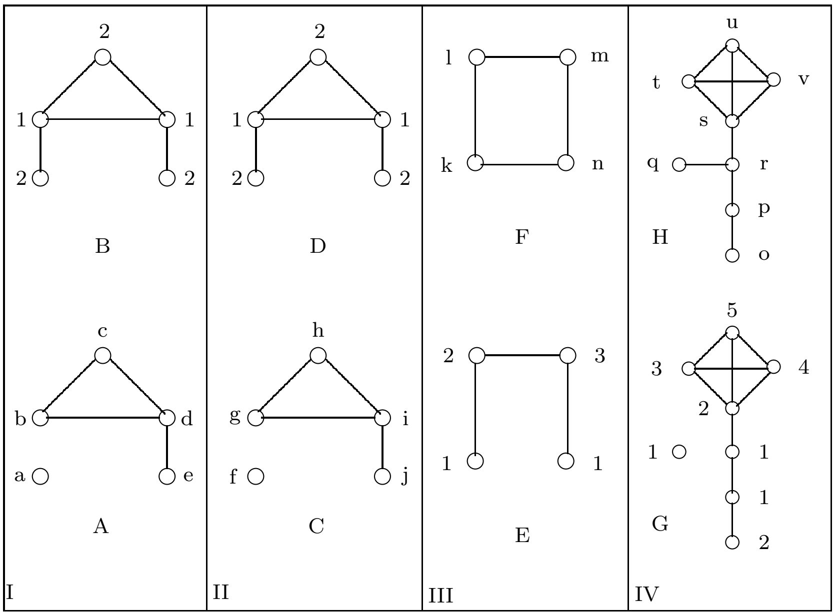



Figure 4: Examples of non-monotonicity. The vertex numbering shows that $\mathrm{ir}^*(B) = 2$, $\gamma^*(D) = 2$, $\beta_0^*(E) = 3$ and $\mathrm{IR}^*(G) = 5$. We prove that $\mathrm{ir}^*(A) = 3$, $\gamma^*(C) = 3$, $\beta_0^*(F) = 2$ and $\mathrm{IR}^*(H) = 4$.

another neighbor in $S'$. Moreover, upon termination of the **while**-loop we know that $S'$ is a minimal dominating set, since there are no vertices triggering the condition in the **while**-loop, so that all vertices in $S'$ have a private neighbor. Equivalently, upon termination the remaining set $S'$ satisfies both Lemma 3 and Lemma 4. □

**Lemma 14** *The iterated irredundance number $ir^*$, the iterated domination number $\gamma^*$, the iterated independence number $\beta_0^*$, and the iterated upper irredundance number $IR^*$ are* **not** *monotone.*

**Proof.** See the graphs in Figure 4 which show the non-monotonicity properties. In the bottom row are subgraphs with a higher iterated numbers than the graphs in the top row. The example for $\mathrm{ir}^*$ in I, for $\gamma^*$ in II, for $\beta_0^*$ in III and for $\mathrm{IR}^*$ in IV.

Since $\mathrm{ir}^*$ and $\gamma^*$ are minimization invariants we give an iterated coloring for the graphs $B$ and $D$, showing that $\mathrm{ir}^*(B) = \gamma^*(D) = 2$. We show that the subgraphs $A$ and $C$ need at least three colors. Any irredundant set in $A$ contains at most one vertex of $b, c, d$, since otherwise not all vertices of the irredundant set will have a private neighbor. After removal of an irredundant set we are therefore left with at least one edge, and indeed $\mathrm{ir}^*(A) = 3$. Since $A$ and $C$ are isomorphic we know that also $\gamma^*(C) \geq \mathrm{ir}^*(C) = \mathrm{ir}^*(A) = 3$, and in fact equality holds.

Since $\beta_0^*$ and IR$^*$ are maximization invariants, we have indicated an iterated coloring for the subgraphs $E$ and $G$, showing that $\beta_0^*(E) = 3$ and IR$^*(G) = 5$. We have $\beta_0^*(F) = 2$, since any independent dominating set in $F$ must include two non-adjacent vertices of the 4-cycle, so that after removal no edges are left. The argument for showing that IR$^*(H) \leq 4$ is slightly longer. Let $S$ be a maximal irredundant set of $H$. If $r \in S$ then $q \notin S$, so $r$ has a private neighbor and at least one of $\{s, t, u, v\}$, say $u$, is in $S$. But after the removal of $S$ we would then be left with isolated vertices, edges and a 3-cycle on $\{s, t, v\}$, and could remove only three more maximal irredundant sets. On the other hand, if $r \notin S$, then $q \in S$ and at least one of $\{s, t, u, v\}$ and one of $\{o, p\}$ is in $S$. After removal of $S$ we are then left with, say, the graph on $\{p, r, s, t, v\}$. In this graph a maximal irredundant set must include one of $\{t, u, v\}$ and one of $\{p, r\}$. After its removal we are left with, say, the 3-path on $\{r, s, t\}$ from which at most two more irredundant sets can be removed. The other possibilities are even easier to argue, and we conclude that IR$^*(H) = 4$. $\square$

# 6 Algorithmic issues

In the first two subsections below we show, respectively, that for a fixed value of $k$, the problems of deciding if a graph $G$ has a *dominating coloring that uses at most $k$ colors, or a *irredundant coloring that uses at most $k$ colors, is polynomial-time solvable for $k < 3$ and NP-complete for $k \geq 3$.

In the final subsection, we show NP-hardness of the problem of finding the *largest* $k$ such that a graph $G$ has a *dominating, a *irredundant, or a *independent coloring, respectively.

## 6.1 Iterated minimal-dominating $k$-colorings

ITERATED MINIMAL-DOMINATING $k$-COLORING (*DOMk)
INSTANCE: Graph $G$.
QUESTION: Does $G$ have an iterated minimal-dominating coloring that uses at most $k$ colors?

It is well known that a graph can be properly colored with two colors if and only if it is bipartite. This is the same as saying that bipartite graphs can be colored using two colors by Algorithm ICA with $P$ = *maximal independent*. Bipartite graphs can also be colored using two colors by Algorithm ICA with $P$ = *minimal dominating*. However, some graphs that are not bipartite can also be 2-colored with $P$ = *minimal dominating*. For example, consider the graph in Figure 3. In this case we will say that a graph is **dominating 2-colorable*, or equivalently, that $\gamma^*(G) = 2$, and such a $P$-coloring is called a **dominating 2-coloring*.

The following polynomial algorithm determines whether or not $\gamma^*(G) = 2$, for any connected graph $G$ of order $n > 1$. For these graphs it is also true that $G$

is *dominating 2-colorable by Algorithm ICA with $P$ = *minimal dominating*. This algorithm makes use of a known polynomial algorithm for solving the following decision problem:

> 2-SATISFIABILITY (2SAT)
> INSTANCE: Collection $C = \{c_1, c_2, \ldots, c_m\}$ of two-literal clauses on a finite set $U$ of variables.
> QUESTION: Is there a truth assignment for $U$ that satisfies all the clauses in $C$?

**Algorithm *dominating-2-colorable**

*Input:* A connected graph $G$.

*Output:* A Boolean variable: *decision* that is **true** if $G$ is 2-$\gamma^*$colorable and **false** otherwise; and a 2-$\gamma^*$coloring of $G$ if $G$ is 2-$\gamma^*$colorable.

1. Color all vertices that are adjacent to endvertices: *blue*.
2. Color all edges between two *blue* vertices: *blue*.
3. Let $B$ equal the set of *blue* edges.
4. **if** the graph $G' = (V, E - B)$ is not bipartite
   (a) **then** /* $G$ is not $2 - \gamma^*$colorable */
   **set** *decision* = **false**;
   **exit**;
   (b) **else** /* $G' = (V, E - B)$ is bipartite */
      i. **for** each connected component $C_i$ of $G'$ **do**
         A. let $\{V_1, V_2\}$ be a proper 2-coloring of $C_i$;
         B. assign the value $c_i$ to each vertex in $V_1$;
         C. assign the value $\overline{c_i}$ to each vertex in $V_2$;
      ii. create a two-literal clause corresponding to each *blue* edge, for example, $(c_i, c_j)$ or $(\overline{c_i}, c_k)$, this creates an instance 2SAT(G) of the 2-satisfiability problem;
      iii. **if** 2SAT(G) is not satisfiable **then** *decision* = **false**; **exit**;
      iv. **else** *decision* = **true**;
      solve the 2SAT(G) problem;
      assign color 1 to all vertices whose corresponding literal is **true**;
      assign color 2 to all vertices whose corresponding literal is **false**;
      **exit**.

The graph in Figure 5 is *dominating 2-colorable. The clauses that result from the *blue* edges are $(\bar{c_1}, \bar{c_2})$ and $(\bar{c_2}, \bar{c_2})$. We can solve the 2-SATISFIABILITY problem by letting $c_1$ be **true** and $c_2$ be **false**. Then all the vertices colored $c_1$ and $\bar{c_2}$ are assigned color 1, while all other vertices are assigned color 2.

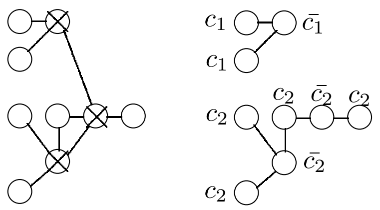



Figure 5: 2-colorable graph ($P$ = minimal dominating).

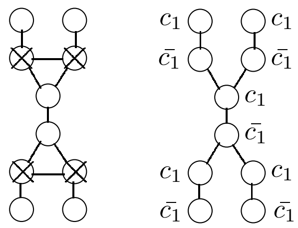



Figure 6: Graph that is not *dominating 2-colorable).

The graph in Figure 6 is not $2 - \gamma^*$colorable, however. The clauses that result from the *blue* edges are $(c_1, c_1)$ and $(\bar{c_1}, \bar{c_1})$. This 2-SATISFIABILITY problem is not solvable.

**Lemma 15** *When* **Algorithm *dominating-2-colorable** *executes instruction 4.(b).iv, it produces a *dominating 2-coloring.*

**Proof.** No two vertices colored 2 are adjacent, that is, the set of vertices colored 2 is independent. Therefore, any vertex colored 2 must be adjacent to a vertex colored 1. If two adjacent vertices are colored 1, they must be connected by a *blue* edge, which means that each of them is adjacent to an endvertex colored 2. □

**Lemma 16** *If a graph $G$ has a *dominating 2-coloring, then* **Algorithm *dominating-2-colorable** *produces a *dominating 2-coloring of $G$.*

**Proof.** Let $G$ be a graph that has a *dominating 2-coloring, and assume that **Algorithm *dominating-2-colorable** cannot 2-color $G$. There are two stages where the algorithm could determine that $G$ is not 2-colorable. The first occurs when all *blue* edges are removed from $G$, and the resulting graph $G' = (V, E - B)$ is not bipartite. If $G'$ is not bipartite, then there is an odd cycle in $G'$, and two adjacent vertices in this cycle must be assigned the same color. These two vertices cannot both be colored 2 because by Lemma 4 they would then have to be adjacent to vertices of a higher color, and there are only two colors. However, they also cannot both be colored 1 because both vertices would need endvertices as private neighbors. If both had adjacent endvertices, then the edge between them would have been colored *blue* by the algorithm.

The second stage where the algorithm could fail to 2-color a graph is when the 2-SATISFIABILITY problem has no solution (i.e. there is no way to assign colors to vertices incident with the *blue* edges, such that at least one of the vertices is colored 1). Note that no two adjacent vertices in $G'$ can be colored the same color. Therefore, if there is a 2-coloring for $G$, then the colors assigned to vertices incident with *blue* edges (1 - **true**, 2 - **false**) should be a satisfiable truth assignment. □

**Theorem 8** *The decision problem *DOMk is polynomial-time solvable for $k = 2$ and NP-complete for any fixed $k$, $k \geq 3$.*

The case $k = 2$ has just been shown. For lack of space the NP-completeness reduction, from the well-known SAT problem, is given in an appendix.

Let us merely mention the following lemmas required for the NP-completeness proof, that may be of independent interest:

**Lemma 17** *Let $f : V \to \{1, 2, \ldots, k\}$ be a *dominating coloring for $G$. Then for any two adjacent vertices $p$ and $q$, if $f(p) = f(q)$, there exists some vertex $w \in N[p] - N[q]$, with $f(w) > f(p)$.*

**Proof.** By Lemma 4, vertex $p$ must be adjacent to a vertex with a higher label that no other vertex with label $f(p)$ is adjacent to. All neighbors of vertex $p$ that are not in $N[p] - N[q]$ are also adjacent to vertex $q$. □

**Lemma 18** *Let $f : V \to \{1, 2, ..., k\}$ be a *dominating $k$-coloring for $G$. Then for any two vertices $p$ and $q$, if $f(p) = f(q)$, then $N[p] - N[q] \neq \emptyset$.*

**Proof.** If $N[p] - N[q] = \emptyset$, then vertices $p$ and $q$ must be adjacent. By Corollary 17, if two adjacent vertices $p$ and $q$ are given the same label, then $N[p] - N[q] \neq \emptyset$. $\Box$

**Lemma 19** *Let $f : V \to \{1, 2, ..., k\}$ be a *dominating $k$-coloring for $G$. Then for any vertex $p$, the degree of $p$ is greater than or equal to $f(p) - 1$.*

**Proof.** This follows from Lemma 3. A vertex $p$ with a label $f(p) > 1$ must be adjacent to vertices with labels $1..p - 1$. $\Box$

## 6.2 Iterated maximal-irredundant $k$-colorings

ITERATED MAXIMAL-IRREDUNDANT $k$-COLORING (*IRRk)
INSTANCE: graph $G = (V, E)$
QUESTION: does $G$ have an iterated maximal-irredundant coloring that uses at most $k$ colors?

**Lemma 20** *If $G$ is 2-colorable with property $P$ = maximal irredundant, then $G$ is also 2-colorable with $P$ = minimal dominating.*

**Proof.** Let $G$ be a graph with $ir^*(G) = 2$, and let $\Pi$ be a *irredundant 2-coloring of $G$. Then $\Pi$ is also a *dominating 2-coloring. Note that a vertex $v$ colored 2 must be adjacent to some other vertex in $G$, as all isolates in $G$ must be colored 1. Vertex $v$ cannot be adjacent to any other vertex colored 2, otherwise $V_2$ is not irredundant. Therefore, vertex $v$ is adjacent to a vertex colored 1. Lemma 3 is satisfied. We do not have to check Lemma 4, since it is also a requirement for *irredundant colorings (see Lemma 5.) $\Box$

**Theorem 9** *The decision problem *IRRk is polynomial-time solvable for $k = 2$ and NP-complete for any fixed $k$, $k \geq 3$.*

The case $k = 2$ has just been shown. The NP-completeness proof is for space reasons to be found in the appendix. We mention an observation used in the proof that may be of independent interest.

**Corollary 1** *Let $f : V \to \{1, 2, ..., k\}$ be a *irredundant $k$-coloring for $G$. Then for any two adjacent vertices $p$ and $q$, if $f(p) = f(q)$, there exists some vertex $w \in N[p] - N[q]$, with $f(w) > f(p)$.*

**Proof.** By Lemma 4, vertex $p$ must be adjacent to a vertex with a higher color that no other vertex with color $f(p)$ is adjacent to. All neighbors of vertex $p$ that are not in $N[p] - N[q]$ are also adjacent to vertex $q$. $\Box$

## 6.3 Iterated Independence, Upper Domination and Upper Irredundance

> ITERATED UPPER IRREDUNDANCE
> INSTANCE: graph $G = (V, E)$
> QUESTION: does $G$ have an iterated maximal-irredundant coloring using at least $k$ colors?

**Theorem 10** *ITERATED UPPER IRREDUNDANCE is NP-complete.*

The proof can be found in the appendix.

> ITERATED MAXIMUM INDEPENDENCE
> INSTANCE: Graph G, positive integer $k$.
> QUESTION: Does $G$ have an iterated maximal-independent coloring that uses at least $k$ colors?

> ITERATED UPPER DOMINATION
> INSTANCE: Graph G, positive integer $k$.
> QUESTION: Does $G$ have an iterated minimal-dominating coloring that uses at least $k$ colors?

**Theorem 11** *ITERATED MAXIMUM INDEPENDENCE and ITERATED UPPER DOMINATION are NP-complete.*

**Proof.** Clearly, the two problems are in $NP$. We use the same construction as for ITERATED MAXIMUM IRREDUNDANCE. As shown in Theorem 10, if the G3C instance is 3-colorable, then $G'$ would have a *independent, and *dominating, $k$-coloring. If $G'$ has an *independent (or *dominating) $k$-coloring, then it has a *irredundant $k$-coloring, and by the theorem, the G3C instance is 3-colorable. □

# 7 Open Problems

Many problems and questions have been raised by our study of iterated colorings of graphs. We conclude by providing a list of some of the most interesting ones.

1. What can you say about $\gamma^*(G)$ and $ir^*(G)$ for planar graphs? Since $i^*(G) = \chi(G)$, we know from the Four Color Theorem [1, 2] that if $G$ is planar, then
$$ir^*(G) \leq \gamma^*(G) \leq i^*(G) = \chi(G) \leq 4.$$
Can you prove that, for planar graphs $G$, $\gamma^*(G) \leq 4$, without using the Four Color Theorem? Failing this, can you prove that $ir^*(G) \leq 4$, for planar graphs $G$, without using the Four Color Theorem?

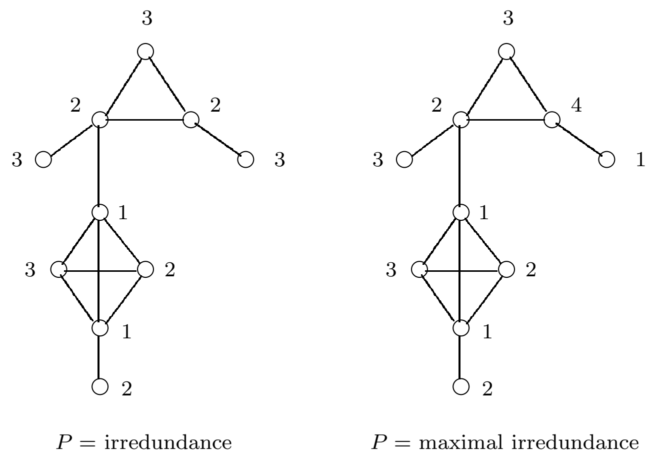



Figure 7: This graph requires 3 colors if $P = $ *irredundant*, but 4 colors if $P = $ *maximal irredundant*.

2. Investigate property $P = $ *irredundant*, rather than $P = $ *maximal irredundant*. The graph in Figure 7 can be 3-colored with $P = $ *irredundant*, but requires four colors if $P = $ *maximal irredundant*.
3. What can you say about iterated coloring numbers for such properties as $P = $ *maximal* 2*-packing* or $P = G[S]$ *is acyclic*.
4. What are the effects of adding or removing edges from $G$ on $ir^*(G)$ and $\gamma^*(G)$. Adding edges to a graph cannot reduce $i^*(G)$, but adding edges can reduce $\mathrm{ir}^*(G)$ or $\gamma^*(G)$. In Figure 7 we see that $\mathrm{ir}^*(G) = 4$, but if we add the new edge shown in Figure 8, $\mathrm{ir}^*(G) = 3$. In Figure 9 we show that $\gamma^*(G)$ can be reduced from 4 to 3 by the addition of a new edge.

# References

[1] K. Appel and W. Haken, Every planar map is four colorable. Part I: Discharging, *Illinois J. Math.* **21**:429-490, 1977.

[2] K. Appel, W. Haken, and J. Koch, Every planar map is four colorable. Part II: Reducibility, *Illinois J. Math.* **21**:491-567. 1977.

[3] T. Beyer, S. M. Hedetniemi and S. T. Hedetniemi, A linear algorithm for the Grundy number of a tree, *Proc. Thirteenth Southeastern Conf. on Combinatorics, Graph Theory and Computing*, *Congr. Numer.* **36**:351-363, 1982.

[4] C. A. Christen and S. M. Selkow, Some perfect coloring properties of graphs, *J. Combin. Theory Ser. B* **27**:49-59, 1979.

[5] E. J. Cockayne, S. T. Hedetniemi and D. J. Miller, Properties of hereditary hypergraphs and middle graphs, *Canad. Math. Bull.* **21**:461-468, 1978.

[6] Michael R. Garey, and David S. Johnson, *Computers and Intractability, A Guide to the Theory of NP-Completeness*, Freeman, New York, 1979.

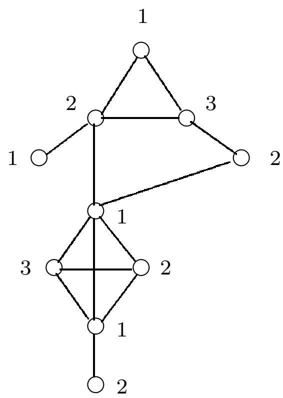



Figure 8: ir$^*(G)$ is reduced from 4 to 3 by the addition of a new edge (see previous figure).

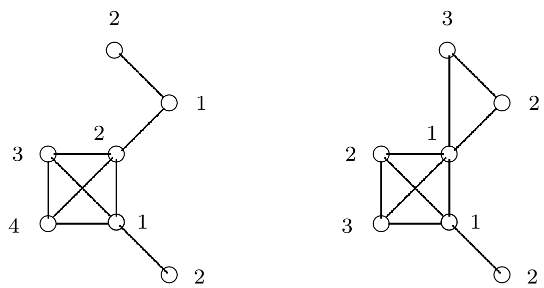



Figure 9: Here we have reduced $\gamma^*(G)$ from 4 to 3 by the addition of a new edge.

[7] N. Goyal and S. Vishwanathan, NP-completeness of undirected Grundy numbering and related problems, preprint, 1997.

[8] P. M. Grundy, Mathematics and games. *Eureka* **2**:6-8, 1939.

[9] T. W. Haynes, S. T. Hedetniemi, and P. J. Slater, *Fundamentals of Domination in Graphs*, Marcel Dekker, New York, 1998, 446 pp.

[10] D. J. Hoffman and P. D. Johnson, Greedy colorings and the Grundy chromatic number of the $n$-cube, *Bull. Inst. Combin. Appl.* **26**:49-57, 1999.

[11] G. Prins, Unpublished manuscript, dated February 8, 1963.

[12] J.A.Telle and A.Proskurowski, Algorithms for Vertex Partitioning Problems on Partial k-Trees, *SIAM Journal on Discrete Mathematics* Vol. 10, No. 4, pp. 529-550, November 1997.

# 8 APPENDIX

We first give the NP-completeness reduction completing the proof of Theorem 8.

**Proof.** Clearly *DOMk is in NP. We need only guess a coloring $\Pi$, and in $O(E)$ time, using Lemmas 3 and 4, it can be verified that $\Pi$ is a *dominating $k$-coloring that uses at most $k$ colors.

A transformation from the well-known SAT problem to *DOMk is given below. We show that given an arbitrary instance of SAT, we can transform the instance into a graph $G$, such that there exists a satisfiable truth assignment for the SAT instance if and only if there exists a *dominating coloring of $G$ with at most $k$ colors, where $k > 2$ is fixed.

Let an arbitrary instance of SAT be represented by a set of variables $U = \{u_1, u_2, u_3, \ldots, u_n\}$ and a set of clauses $C = \{c_1, c_2, c_3, \ldots, c_m\}$. Our transformation has three components: variable components, clause components, and a communication component.

1. Variable components: For each variable $u_i$ in the SAT instance, create a variable component. Start with a complete graph $K_k$ on $k$ vertices and identify any two vertices in $K_k$, labeling them $p_i$ and $\bar{p_i}$. Create two new vertices labeled $u_i$ and $\bar{u_i}$. Connect $u_i$ to all vertices in $K_k$ except for $p_i$, and connect $\bar{u_i}$ to all vertices in $K_k$, except for $\bar{p_i}$
2. Clause components: For each clause $c_j$ in the SAT instance, create a clause component, consisting of a path on three vertices, with the first vertex in the path labeled $c_j$. Connect each $c_j$ vertex to the $u$ variable vertices whose names correspond to the literals in the clause $c_j$.
3. Communication component: Create a complete graph $K_{2k-2}$ on $2k-2$ vertices, and label $k-1$ of the vertices with labels $r_1..r_{k-1}$. Add $k-1$ additional vertices $s_1..s_{k-1}$ along with the edges $(r_i, s_i)$, $1 \le i \le k-1$. Add two more vertices $x_1$ and $x_2$ and the two edges $(x_1, s_1)$ and $(x_1, x_2)$. Form a complete graph among the $s$ vertices and the $c$ vertices from the clause components.

Clearly, this construction is polynomial with respect to the size of the SAT input. The remainder of the proof requires two parts. First, we show that if the SAT instance has a satisfiable truth assignment, then $G$ has a *dominating coloring that uses at most $k$ colors. Given a satisfiable truth assignment $g$, an iterated dominating coloring function $f$ can be found in this manner:

1. Variable components: if SAT variable $u_i$ is assigned **true**, then $f(u_i) = f(p_i) = 1$ and $f(\bar{u_i}) = f(\bar{p_i}) = 3$; otherwise if SAT variable $u_i$ is assigned **false**, then $f(u_i) = f(p_i) = 3$ and $f(\bar{u_i}) = f(\bar{p_i}) = 1$. Assign the colors $2, 4, ..., k$ to the other $k-2$ vertices in the variable component using a different label for each of these vertices.
2. Clause components: assign all $c_j$ variables the color 2, and the middle vertices in the component paths the color 3, and assign the color 1 to all the end-vertices in the paths.

3. Communication component: Assign all $r$ vertices the color 1, and assign to the other $k-1$ vertices in the complete graph the colors $2, 3, 4, 5, \ldots, k$, using a different label for each of these vertices. Let $f(x_1) = 3$ and $f(x_2) = 1$. Let $f(s_i) = i + 1$, for $1 \leq i \leq k-1$.

We consider each vertex in turn to show that this assignment is a valid *dominating coloring.

1. No two adjacent vertices in a variable component have the same color, and every vertex is in a complete graph with $k$ vertices, all which have different colors. Therefore, every vertex $i$ with color $f(i) > 1$ is adjacent to some vertex with color $f(j)$, for any $j$, $1 \leq j < i$. Note also that no $u_i$ or $\bar{u}_i$ variable has the same color as a $c_i$ variable.
2. In the clause components, no vertices, other than the $c_j$ vertices are adjacent to any vertices with the same color. The vertices colored 3 are adjacent to vertices colored 1 and 2. All $c_j$ vertices are colored 2, and they are adjacent to a $u$ or $\bar{u}$ vertex that is colored 1; note this $u$ or $\bar{u}$ vertex corresponds to a true literal from the SAT instance. Each $c_j$ vertex is adjacent to a vertex colored 3 (the vertex in the middle of the path) that no other vertex colored 2 is adjacent to.
3. In the communication component, the $x_1$ and $x_2$ variables are not adjacent to any vertices that match their own color. Furthermore, $x_1$ is adjacent to a vertex colored 1 and a vertex colored 2. Other than $s_1$, no $s_i$ variable is adjacent to a variable with the same color as its own, and each $s_i$ is adjacent to some vertex of every other color between 1 and $k$ inclusive. Vertex $s_1$ is the only vertex with color 2 adjacent to $x_1$, so it can get a private neighbor from $x_1$. Finally, consider the vertices in the $2k-2$ clique. The $r$ vertices have private neighbors of the corresponding $s$ vertices. The other vertices in the clique are not adjacent to any other vertex with the same color, but they are adjacent to all colors other than their own.

Second, we must show that if $G$ has a *dominating coloring $f$ using $k$ or fewer colors, then the SAT instance is satisfiable. Given a *dominating coloring for $G$ with $k$ or fewer colors, a truth assignment $g$ can be found for the SAT instance in this manner: if $f(u_i) = 1$, then $g(u_i) = \textbf{true}$, otherwise $g(u_i) = \textbf{false}$. We show this is a valid truth assignment by proving that each $c_j$ vertex is adjacent to a $u$-vertex with a **true** label 1. We also show that it is not possible that $f(u_i) = f(\bar{u}_i) = 1$. This requires several observations.

1. Consider the vertices in the $2k-2$ clique in the communication component. The $k-1$ unlabeled vertices all share the same closed neighborhoods, so by Corollary 18, they must all be different colors. Also, no unlabeled vertex can have the same color as any one of the $r$ vertices because the closed neighborhood of any unlabeled vertex is a subset of the closed neighborhood of any $r$ vertex. Since only $k$ colors are used for $G$, it follows that all $r$ vertices are colored the same, and that they must get private

neighbors from the $s$-vertices. Since the $s$-vertices are private neighbors, each must have a color greater than 1.

2. Two vertices $s_i$ and $s_j$, $i < j$, cannot have the same color because $N[j] - N[i] = \{r_j\}$, and $r_j$ cannot be a private neighbor for $s_j$, since $s_j$ is a private neighbor for $r_j$. Therefore all $s$-vertices must have different colors other than 1, necessarily $2..k$. These vertices are private neighbors for $r$ vertices which all must have been colored 1.

3. We now show that the $c$ vertices must all be colored 2. Note that the $c$ vertices must have the same color as $s_1$ :

   (a) vertex $c_j$ cannot have a color of 1, because the $s$ vertices are private neighbors of the $r$ vertices that are colored 1.

   (b) vertex $c_j$ cannot have the same color as any of the $s_i$ vertices $i > 1$, since $N[s_i] - N[c_j] = \{r_i\}$ and $r_i$ cannot be a private neighbor for $s_i$.

   (c) There is only one color possible for $c_j$ and that is $s_1$'s color. Vertex $s_1$ will get its private neighbor from $x_1$ and $x_1$ must be colored 3: vertex $x_1$ cannot be colored lower than 3 and be a private neighbor for $s_1$, and it cannot have a higher color than 3 since it is adjacent to only two vertices.

4. In any vertex component, either the $k$-clique is colored with $k$ colors or the $k$-clique is colored with $k - 1$ colors and the two labeled vertices are colored the same color (this is because all vertices in the clique have neighborhoods that are contained in the neighborhood of any unlabeled vertex). If the $k$-clique is colored with $k-1$ colors then the two $u$-vertices must be private neighbors for the labeled vertices, and their color must be the same as the color missing in the clique.

5. The two $u$-vertices in a vertex component must not both be assigned 1; otherwise the $k$-clique would have to be colored with $k$ colors, and the $k$-clique vertex colored 1 would have no private neighbor.

6. If a $u$-vertex is assigned a color greater than 1, then the $k - 1$ vertex component clique vertices that it is joined to must be all different colors, and these colors must be different from $u$, otherwise either the $u$ vertex or the two clique vertices would not have private neighbors in the case of a match.

7. No $c_i$ clause vertex can use a $u$-vertex as a private neighbor, because if the $u$-vertex is assigned a color other than 1, then it is adjacent to a vertex colored 2 in its component's clique. Therefore all $c_i$ vertices get their private neighbors from the center vertices in the paths. These center vertices are colored 3. Each $c_i$ clause vertex must be adjacent to a vertex colored 1. These must come from the $u$-vertices.

□

We now give the NP-completeness reduction completing the proof of Theorem 9.

**Proof.** Clearly *IRRk is in $NP$. We need only guess a coloring $f$, and in $O(E)$ time, using Lemmas 5 and 6, it can be verified whether $f$ is a *irredundant coloring that uses at most $k$ colors.

A transformation from the Exact Cover by 3-Sets problem (X3C) from [6] to *IRRk is given below. We show that given an arbitrary instance of X3C, we can transform the instance into a graph $G$, such that there exists a satisfiable truth assignment for the X3C instance if and only if there exists an iterated irredundance coloring of $G$ with at most $k$ colors, where $k > 2$ is fixed.

> EXACT COVER BY 3-SETS
> INSTANCE: Set $X$ of elements, $|X| = 3q$, and a collection $C$ of 3-element subsets of $X$.
> QUESTION: Does $C$ have an exact cover $C'$; i.e., every element of $X$ appears in exactly one subset in $C'$?

Let a set of elements $X = \{x_1, x_2, x_3, ..., x_{3q}\}$ and a collection of subsets $C = \{c_1, c_2, c_3, ..., c_m\}$ be an arbitrary instance of X3C. Our transformation has two types of components: element components and subset components.

1. Element components: For each element $x_i$ in the X3C instance, create an element component. Start with a complete graph on $k+1$ vertices and select any two vertices in the complete graph, labeling them $x_i$ and $y_i$. Create another vertex labeled $p_i$, and add the edge $\{p_i, y_i\}$.
2. Subset components: For each subset $c_j$ in the X3C instance, create a subset component. For the subset components, again begin with a complete graph on $k+1$ vertices, and select any three vertices in the complete graph labeling them $c_{j,1}$, $c_{j,2}$, and $c_{j,3}$/ Create three new vertices $q_{j,1}$, $q_{j,2}$, $q_{j,3}$ and add edges $\{q_{j,1}, c_{j,1}\}$, $\{q_{j,2}, c_{j,2}\}$ and $\{q_{j,3}, c_{j,3}\}$. Connect $c_{j,1}$, $c_{j,2}$, and $c_{j,3}$ to the $x_i$ element vertices whose names correspond to the elements in the subset $c_j$.

Clearly, this construction is polynomial with respect to the size of the X3C input. Figure 10 illustrates the construction described above for $k = 4$.

In this part of the proof, we will use the notation $c_{j*}$ to refer to an arbitrary vertex from $c_{j1}$, $c_{j2}$, or $c_{j3}$. Similarly, the notation $q_{j*}$ will be used. Also, the term *xvertex* will refer to any vertex labeled $x_i$. First we show that if the X3C instance has an exact cover, then $G$ has a *irredundant coloring that uses at most $k$ colors. Given an exact cover $C'$, a *irredundant coloring $f$ can be found in this manner:

1. Subset components: If $c_j$ is a subset in the exact cover, let $f(c_{j1}) = f(c_{j2}) = f(c_{j3}) = 2$, and $f(q_{j1}) = f(q_{j2}) = f(q_{j3}) = 3$, and assign the colors $3, 4, \ldots, k$ to the other $k-2$ vertices in the subset component using a different color for each of these vertices. If $c_j$ is not a subset in the exact cover, let $f(c_{j1}) = f(c_{j2}) = f(c_{j3}) = 1$, and $f(q_{j1}) = f(q_{j2}) = f(q_{j3}) = 2$, and assign the colors $2, 3, \ldots, k-1$ to the other $k-2$ vertices in the

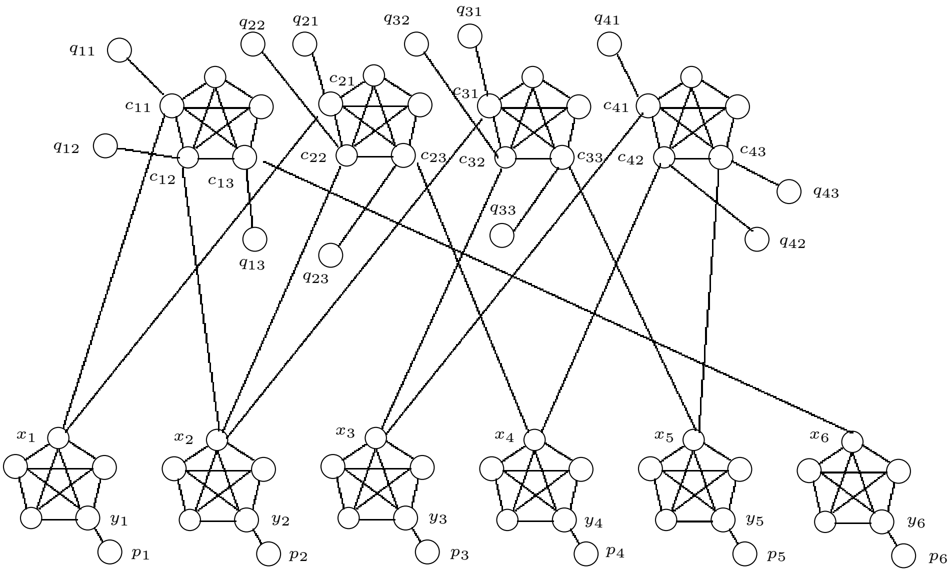



Figure 10: Transformation to *IRRk from X3C where $X = \{x_1, x_2, \ldots, x_6\}$ and $C = \{\{x_1, x_2, x_6\}, \{x_1, x_2, x_4\}, \{x_2, x_3, x_5\}, \{x_3, x_4, x_5\}\}$.

complete graph in the subset component using a different color for each of these vertices.

2. Element components: For each element component assign $x_i$ and $y_i$ the value 1, and $p_i$ the value 2. Assign $2, 3, 4, \ldots, k$ to the other $k-1$ vertices in the complete graph.

We consider each vertex in turn to show that this assignment is a valid *irredundant coloring.

1. The only two adjacent vertices in an element component that have the same color are $x_i$ and $y_i$. The $y_i$ vertices will have the $p_i$ vertices as private neighbors, and the $x_i$ vertices will have private neighbors in the $c_j$ component that covers it from the exact cover. No vertex in an element component can be given a lower value because it would not have a private neighbor with respect to that value.

2. In the subset component, all vertices have private neighbors within the subset component. In the components where the corresponding subsets form the exact cover, the $c_{j,*}$ vertices are being used as private neighbors by some $x$ vertices, and so no vertices in these subset components can be colored 1. No vertex can be given a lower color greater than 1 because it would not have a private neighbor with respect to that color.

Now we must show that if $G$ has a *irredundant coloring $f$ using $k$ or fewer colors that the X3C instance has an exact cover. Given a *irredundant coloring for $G$ with $k$ or fewer colors, an exact cover $C'$ can be found for the X3C instance in this manner: if $f(q_{1j}) > 2$, then $c_j$ is in $C'$; otherwise $c_j$ is not in $C'$. We show this is a valid exact cover by proving that each of the $x_i$ vertices is colored 1 and must get its private neighbor from a $c_{j*}$ vertex. Further we show that if a $c_{j*}$ vertex is used as a private neighbor, then $c_{j1}$, $c_{j2}$, and $c_{j3}$ must all be used as sole private neighbors for some $x$ vertices. This is the only case in which $f(q_{1j}) > 2$.

This requires several observations.

1. Consider the vertices in the $k+1$ clique in an element component. To color these vertices with $k$ colors, two vertices must be the same color, and the private neighbors must come from outside the clique. Therefore the two vertices with the same color are $x_i$ and $y_i$. Every other vertex inside the clique will use itself as a private neighbor with respect to its color set. The vertex $p_i$ is a private neighbor for $y_i$, so clearly $f(p_i) > 1$. By Lemma 2, if $f(p_i) > 1$, then either $p(i)$ would have to be adjacent to a vertex colored 1, or it would have to destroy a private neighbor set for a vertex colored 1. This can only occur if $f(y_i) = 1$.

2. Consider a vertex $c_{j*}$ that is used as a private neighbor for some vertex $x_i$. We show that all three of $c_{j1}$, $c_{j2}$, and $c_{j3}$ must be used as sole private neighbors for some $x$ vertices.

   First, no vertex in the $c_j$ clique can be colored 1, otherwise $c_{j,*}$ could not be used as a private neighbor for the $x_i$ vertex. It follows that the $k+1$ vertices in the subset clique are assigned from the $k-1$ colors $\{2, 3, 4, \ldots, k\}$. The $k-2$ unlabeled vertices in the clique must all have different colors, and these must be colored differently than the labeled vertices. Therefore, all the labeled vertices in the $c_j$ component must be the same color. The labeled clique vertices must have private neighbors from the $q$ vertices. Note then that the $q$ vertices in this component must have colors greater than 2. By Lemma 6, we know that a vertex colored greater than 2 must either have no private neighbor with respect to $V_1$ or must be adjacent to all private neighbors of some vertex in $V_1$. A $q$ vertex in this component is not adjacent to any vertex with value one, so it must destroy the private neighbor set of some vertex with color 1, namely an $x$ vertex. Therefore it must be that $c_{j1}$, $c_{j2}$, and $c_{j3}$ are all needed as a sole private neighbor for some $x$ vertex.

3. Using the reasoning from above, if $f(q_{j1} > 2$, then either $c_{j1} = 1$ or vertex $c_{j1}$ is used as a unique private neighbor for an $x_i$ vertex. If $f(c_{j1}) = 1$, then $q_{j1}$ is an isolate in $G_2$ and must be colored 2, a contradiction. Therefore, if $f(q_{j1}) > 2$, then $c_{j1}$, $c_{j2}$, and $c_{j3}$ must all be used as sole private neighbors for $x$ vertices.

4. We have shown that each of the $x$ vertices (and there are $3q$ of these vertices ) must have a private neighbor from a $c$ vertex. These private neighbors must be sole private neighbors, and must be from a subset

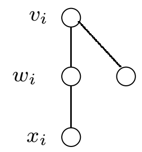



Figure 11: Vertex-edge component.

component where all three $c$ vertices are used as sole private neighbors. Therefore there are exactly $q$ subset components with $f(q_{j1}) > 2$, and the corresponding subsets form an exact cover $C'$.

□

We give the proof of Theorem 10:

**Proof.** Clearly ITERATED UPPER IRREDUNDANCE is in $NP$. We need only guess a coloring $f$, and in polynomial time verify that $f$ is a *irredundant coloring that uses at least $k$ colors.

Given an arbitrary instance $G$ of Graph 3-Colorability, we transform the instance into a graph $G'$ and positive integer $k$, such that there exists a (proper) 3-coloring for $G$ if and only if there exists a *irredundant $k$-coloring for $G'$.

First, create a graph $G''$ by adding a disjoint $K_2$ to $G$. Label the $K_2$ vertices with $v_{|V|+1}$ and $v_{|V|+2}$ and the edge with $e_{|E|+1}$. We will transform $G''$ to a graph $G'$ that has two types of components: vertex-edge components and edge components.

1. Vertex-edge components: For each vertex $v_i$ in $G''$ create a tree as shown in Figure 11, rooting the tree at $v_i$. For each edge $e_j$ in graph $G''$, add the same tree as shown in Figure 11, rooting the tree at $v_{|V''|+j}$.
2. Edge components: For each edge $e_j$ in $G''$, create an additional vertex $e_j$. Three edges will be added from this vertex to the vertex-edge components: add edges from this vertex to the two $v$-vertices that are endpoints of the edge and also to the $v_{|V''|+j}$ vertex from the vertex-edge component. Form a clique among the $e_j$ vertices.
3. Let $k = |E''| + 3$.

Clearly, this construction is polynomial with respect to the size of the G3C input.

First we show that if $G$ has a 3-coloring, then $G'$ has a *irredundant coloring that uses $k$ colors. Given a 3-coloring $g$ for $G$, an iterated irredundance coloring function $f$ can be found in this manner:

1. Vertex-edge components: Let $f(v_i) = g(v_i)$, for $i \leq |V|$. Let $f(v_{|V|+1}) = 1$ and $f(v_{|V|+2}) = 2$. For each $v_i$ vertex, $i = |V''| + j$, representing an edge $e_j$ with endpoints $v_a$ and $v_b$ in $G''$, let $f(v_{|V|+j}) = 6 - g(v_a) - g(v_b)$; in

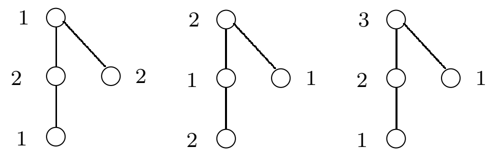


Figure 12: Coloring a vertex-edge component.

other words either 1, 2, or 3, whichever of those colors is not used for its endpoints $v_a$ and $v_b$. Color the other vertices in the vertex-edge trees corresponding to the tree in Figure 12 that has the same root color as $f(v_i)$.

2. Edge components: Let $f(e_j) = 3 + j$.

It is easy to see that this assignment is a valid *irredundant $k$-coloring. Each color class $C_i$ is a maximal independent set (and therefore maximal irredundant) with respect to the graph $G'_i$. Note that each $e_j$ vertex is adjacent to vertices colored 1, 2, and 3 from the vertex-edge component.

Now we must show that if $G'$ has an iterated irredundance coloring $f$ using $k$ or more colors, then the G3C instance has a proper 3-coloring. Given a *irredundant coloring $f$ for $G'$ with $k$ or more colors, a proper 3-coloring $g$ for $G$ can be found by letting $g(v_i) = f(v_i)$ for each vertex $v_i$ in $G$. We will show that this is a proper 3-coloring.

This requires several observations.

1. In each vertex-edge component tree, at least one of the vertices $v_i$, $w_i$, or $x_i$ must be colored 1. Otherwise $C_1$ is not a maximal irredundant set.
2. If two adjacent vertices in a vertex-edge component tree are the same color, then the two vertices must be $v_i$ and $w_i$. In this case, $v_i$ and $w_i$ must be colored 1, and both can get private neighbors from within the tree. No vertex outside the clique will have its private neighborhood set contained within the clique.

3. Every vertex $p$ in the clique must be adjacent to vertices colored $m$, for $1 \leq m < f(p)$. Otherwise $C_m \cup p$ is irredundant in $G_m$, contradicting the maximality of $C_m$.
4. Since there are $k - 3$ vertices in the clique, and $k$ colors used to color $G'$, there are at least three colors missing from the clique. Consider the largest color $m$ missing in the clique, and let $y$ be a vertex with $f(y) = m$. Then for every other missing color in the clique, $y$ must be adjacent to some vertex of that color, otherwise $y$ could be given a lower color. Since no non-clique vertex is adjacent to more than two other non-clique vertices, there are only three colors missing from the clique, and all vertices in the clique have a different color.

5. Let $c_1 \leq c_2 \leq c_3$ be the three colors missing from the clique. Let $y$ be any vertex not in the clique. Then if $f(y) > c_1$, then $y$ must be adjacent to a vertex colored $c_1$, if $f(y) > c_2$, then $y$ must be adjacent to a vertex colored $c_2$, and if $f(y) > c_3$, then $y$ must be adjacent to a vertex colored $c_3$. Note that no vertex outside the clique can have a value greater than $c_3$, because its degree outside the clique is at most 2.

6. Consider a vertex $y$ with value $c_3$. It is adjacent to two vertices outside the clique, so $y$ must either be a $v$ or $w$ vertex from a vertex-edge component. If $y$ is a $v$ vertex, then the $w$ vertex is colored $c_2$ and the other two vertices in the tree component are colored $c_1$. If $y$ is a $w$ vertex, then the $v$ vertex is colored $c_2$. In either case, $c_1 = 1$.

7. Let $y$ be a vertex with $f(y) = c_3$ in a vertex-edge component with root $v_i$ (possibly $v_i = y$). Let $p$ be any clique vertex that is not adjacent to $v_i$. Then, $f(p) > f(w_i)$; otherwise $C_{f(p)} \cup w_i$ would be irredundant in $G_p$. Therefore, $f(p) > c_2$. Further, $f(p) > c_3$, otherwise $C_{f(p)} \cup y$ would be irredundant in $G_p$. Note that $p$ cannot be adjacent to another vertex colored $p$ (all clique vertices are different colors, and an adjacent $v_j$ vertex colored $f(p)$ would have to be adjacent to a $c_1$ and $c_2$ by (4) above and could not get a private neighbor with respect to $C_{f(p)}$.

8. Let $p$ be any clique vertex. If $f(p) < c_3$, then $p$ must be adjacent to all vertex-edge components that contain a vertex colored $c_3$. Also, if $f(p) > c_3$, then $p$ must be adjacent to some vertex-edge component colored $c_3$. Consider the clique vertex that represents the $K_2$ edge added to make $G''$. Its neighborhood outside the clique is disjoint from any other clique vertex. Therefore, no clique vertex can be adjacent to all vertex-edge components containing $c_3$ vertices. The clique vertices are colored $4..|E''|$, and each must be adjacent to a vertex colored 1, 2, and 3.

9. The function $g$ will employ only the colors 1, 2, and 3. Also, any two endpoints of an edge in $G$ will be colored differently.

□