Abstract

This paper studies degree-bounded vertex partitions, derives analogues for well-known results on the chromatic number and graph perfection, and presents two algorithms for constructing degree-bounded vertex partitions. The first algorithm minimizes the number of partition classes. The second algorithm minimizes a weighted sum of the partition classes where the weight of a partition class depends on the level of adjacency among its vertices.

1 Introduction

A coloring partitions the vertex set of a graph $G = (V, E)$ into subsets of pairwise non-adjacent vertices. A classical problem in combinatorial optimization is to find a coloring which uses the smallest possible number of color classes. The minimum number of color classes required is known as the chromatic number $\chi(G)$. If $V$ represents a set of objects and $E$ the set of conflicting pairs, graph coloring solves the problem of dividing $V$ into the minimum number of conflict-free subgroups.

A second application of graph coloring arises from its relation to another classical problem in combinatorial optimization. The maximum clique problem asks for the largest subset of pairwise adjacent vertices in a graph. Since the color classes of a coloring are edgeless, a subset of pairwise adjacent vertices meets each color class at most once. Consequently, $\chi(G)$ is an upper bound on the cardinality of a maximum clique, and researchers [2, 21, 23] use graph coloring in branch and bound solvers for the maximum clique problem.
Generalized graph coloring describes the partitioning of the vertices into classes whose induced subgraphs satisfy particular constraints [22]. For example, $k$-improper colorings have the property that each color class induces a subgraph of maximum degree at most $k$ [1, 7, 12]. The generalization to $k$-improper colorings suggests two optimization problems.

The first seeks to partition a graph into degree-bounded subgraphs using the smallest possible number of partition classes. This solves the problem of dividing $V$ into the minimum number of subgroups such that each vertex has a bounded number of conflicts in its partition class. The second problem minimizes a weighted sum of the partition classes. In this case, the weight of a partition class depends on the level of adjacency among its vertices. This second problem produces a bound on the cardinality of subgraphs defined as degree-based clique relaxations [18] and leads to a generalization of graph perfection.

The $k$-improper chromatic number is associated with the first optimization problem and has been studied in a variety of contexts [6, 11, 13]. Much of this research focuses on random graphs and generalizations of the Four Color Theorem. Some applications of $k$-improper coloring include radio-frequency assignment [11] and network security [19]. The second optimization problem appears to be new.

The remainder of this paper is organized as follows. Section 2 discusses some relevant definitions and notation. Section 3 explores the relationship between cohesive subgraphs and degree-bounded vertex partitions. Section 4.1 adapts a well-known graph coloring algorithm [15] to solve the problem of minimizing the number of partition sets. Section 4.2 uses the algorithm from Section 4.1 to solve a more general class of problems. Section 5 summarizes and suggests some future research directions.

2 Preliminaries

All graphs $G = (V, E)$ in this paper are finite, undirected, and simple. The girth, $g(G)$, is the length of the smallest cycle in $G$. Given a vertex $v \in V$, define $N_G(v) := \{u \in V \mid uv \in E\}$,
\[ \deg_G(v) := |N_G(v)|, \ \Delta(G) := \max_{v \in V} \deg_G(v), \ \text{and} \ \delta(G) := \min_{v \in V} \deg_G(v). \] Let \(G[K]\) denote the subgraph induced by \(K \subseteq V\). In this paper, \(k \geq 1\) is always a positive integer.

**Definition 1.** \(K \subseteq V\) induces a \(k\)-plex if \(\delta(G[K]) \geq |K| - k\).

**Definition 2.** \(C \subseteq V\) induces a co-\(k\)-plex if \(\Delta(G[C]) \leq k - 1\).

**Definition 3.** A partition of the vertex set into disjoint, nonempty co-\(k\)-plexes defines a co-\(k\)-plex coloring of \(G\).

Definitions 1 and 2 are due to Seidman and Foster [20]. Co-\(k\)-plexes are also known as \((k - 1)\)-dependent or \((k - 1)\)-stable sets [11]. Notice that 1-plexes and co-1-plexes are complete subgraphs and stable sets, respectively. Let \(\omega_k(G)\) denote the cardinality of a largest \(k\)-plex in \(G\), \(\alpha_k(G)\) the cardinality of a largest co-\(k\)-plex in \(G\), and \(\Pi\) the set of all co-\(k\)-plex colorings of \(G\).

**Definition 4.** The co-\(k\)-plex chromatic number of \(G\) is defined as

\[
\chi_k(G) := \min \{ \sum_{C \in P} \omega_k(G[C]) : P \in \Pi \}. 
\]

**Definition 5.** The cardinality co-\(k\)-plex chromatic number of \(G\) is defined as

\[
\bar{\chi}_k(G) := \min \{ m : \exists P \in \Pi \text{ s.t. } |P| = m \}. 
\]

\(\bar{\chi}_k(G)\) is exactly the \((k - 1)\)-improper chromatic number [22]. A \(\bar{\chi}_k\)-optimal coloring partitions \(V\) using the smallest possible number of co-\(k\)-plex sets. A \(\chi_k\)-optimal coloring \(C_1, ..., C_m\) satisfies \(\chi_k(G) = \sum_{i=1}^{m} \omega_k(G[C_i])\) and thus

\[
\bar{\chi}_k(G) \leq m \leq \sum_{i=1}^{m} \omega_k(G[C_i]) = \chi_k(G). 
\]

Notice also that \(\chi_1(G) = \bar{\chi}_1(G) = \chi(G)\). Moreover, a coloring is \(\chi_1\)-optimal if and only if it is \(\bar{\chi}_1\)-optimal. However, this relationship fails for \(k > 1\). To see this, consider the trivial
example of $k$ pairwise non-adjacent vertices. The unique $\chi_k$-optimal coloring consists of a single color class. On the other hand, assigning each vertex to a distinct color class defines a $\chi_k$-optimal coloring which uses $k$ color classes.

A co-$k$-plex $C$ is called deficient whenever $|C| < k$. A deficient co-$k$-plex $C$ satisfies $\omega_k(G[C]) = |C|$. A compact co-$k$-plex coloring has at most one deficient co-$k$-plex set.

**Lemma 1.** Every co-$k$-plex coloring $C_1, \ldots, C_m$ can be changed into a compact co-$k$-plex coloring $C'_1, \ldots, C'_p$ such that $p \leq m$ and $\sum_{i=1}^{m} \omega_k(G[C_i]) = \sum_{i=1}^{p} \omega_k(G[C'_i])$.

**Proof.** Consider the co-$k$-plex coloring $C_1, \ldots, C_m$. Suppose there are two deficient co-$k$-plexes $C_i$ and $C_j$. It follows that $\omega_k(G[C_i]) + \omega_k(G[C_j]) = |C_i| + |C_j|$. Choose a vertex $v \in C_j$. Define $C'_j := C_j \setminus \{v\}$ and $C'_i := C_i \cup \{v\}$. Now $|C_i \cup \{v\}| \leq k$ ensures that $C'_i$ and $C'_j$ both remain co-$k$-plexes. Moreover,

$$\omega_k(G[C'_i]) + \omega_k(G[C'_j]) = (|C_i| + 1) + (|C_j| - 1) = \omega_k(G[C_i]) + \omega_k(G[C_j]).$$

Continue moving vertices from $C_j$ to $C_i$ until either $C'_j = \emptyset$ or $|C'_i| = k$, in which case the number of deficient sets has been reduced. This procedure can be repeated until the co-$k$-plex coloring $C'_1, \ldots, C'_p$ is compact. It is also clear that $p \leq m$ since the procedure can only reduce the number of partition sets in the co-$k$-plex coloring. \qed

### 3 Bounding cohesive subgraphs

This section analyzes the relationship between $\chi_k(G)$ and $\omega_k(G)$. Section 3.1 introduces the notion of $k$-plex perfection, offers some examples of $k$-plex perfect graphs, and explores $k$-plex analogues for certain properties of perfection. Section 3.2 discusses a theorem of Erdős [10].
### 3.1 k-plex perfection

A coloring function partitions $V$ into co-1-plexes to obtain an upper bound on $\omega_1(G)$. Similarly, partitioning $V$ into degree-bounded subgraphs leads to an upper bound on $\omega_k(G)$. Let $S_1, ..., S_m$ be a co-$k$-plex coloring of $G$, and let $K \subseteq V$ be a maximum $k$-plex in $G$. Observe that

$$\omega_k(G) = |K| = \sum_{i=1}^{m} |K \cap S_i| \leq \sum_{i=1}^{m} \omega_k(G[S_i]),$$

where the inequality follows from the fact that $k$-plexes are closed under set inclusion [20]. Notice that $\chi_k(G) \geq \omega_k(G)$. Recall that a graph $G$ is perfect if $\chi(G') = \omega_1(G')$ for every vertex-induced subgraph $G' \subseteq G$.

**Definition 6.** A $k$-plex perfect graph $G$ satisfies $\omega_k(G') = \chi_k(G')$ for all vertex-induced subgraphs $G' \subseteq G$.

For example, a co-$k$-plex $S$ satisfies $\chi_k(S) = \omega_k(S)$ by definition. Therefore, co-$k$-plexes are $k$-plex perfect because every vertex-induced subgraph of a co-$k$-plex is also a co-$k$-plex [20]. Recall that a finite set $X$ and a family $\mathcal{I}$ of subsets of $X$ define a matroid if the following axioms hold:

1. $\emptyset \in \mathcal{I}$

2. $I' \subseteq I \in \mathcal{I}$ implies $I' \in \mathcal{I}$

3. Every maximal set in $\mathcal{I}$ has the same cardinality

Given a graph $G = (V, E)$, define

$$\mathcal{K} = \{K \subseteq V : \delta(G[K]) \geq |K| - k\}.$$

$\mathcal{K}$ is the set of $k$-plexes in $G$, and $(V, \mathcal{K})$ satisfies the first two matroid axioms for any graph.

**Theorem 1.** If $M := (V, \mathcal{K})$ defines a matroid, then $G$ is $k$-plex perfect.
Proof. Given any vertex-induced subgraph \( G' = (V', E') \), define \( D := V \setminus V' \) and \( K' = \{ K \subseteq V' : \delta(G[K]) \geq |K| - k \} \). Observe that

\[
(V', K') = (V \setminus D, K') =: M \setminus D
\]

is again a matroid known as a deletion matroid, so it suffices to show \( \chi_k(G) = \omega_k(G) \).

Define \( x(A) = \sum_{a \in A} x_a, S = \{ S \subseteq V : \Delta(G[S]) \leq k - 1 \} \), and \( S_v = \{ S \in S : v \in S \} \). Consider the following dual pair of linear programs:

\[
\begin{align*}
\max \{ x(V) : x \geq 0, x(S) \leq \omega_k(G[S]) \text{ for all } S \in S \} & \quad (1) \\
\min \{ \sum_{S \in S} \omega_k(G[S])y_S : y \geq 0, y(S_v) \geq 1 \text{ for all } v \in V \} & \quad (2)
\end{align*}
\]

Since \( M \) is a matroid, a theorem of Edmonds [9] implies that optimal solutions for (1) and (2) are integral. Observe that \( \omega_k(G) \) and \( \chi_k(G) \) are the optimal objective values for (1) and (2), respectively. Moreover, \( \omega_k(G) = \chi_k(G) \) by strong duality. \( \square \)

**Corollary 1.** If \( G \) is a \( k \)-plex, then \( G \) is \( k \)-plex perfect.

**Proof.** Given any \( K' \subset V \) and \( v \in V \setminus K' \), \( K' \cup \{ v \} \) defines a \( k \)-plex. It follows that all maximal \( k \)-plexes have cardinality \( \omega_k(G) = |V| \), so \( G \) is \( k \)-plex perfect by Theorem 1. \( \square \)

Recall that an \( r \)-partite graph is \( r \)-colorable. The complete \( r \)-partite graphs have all possible edges between distinct color classes.

**Theorem 2.** If \( G \) is the complete \( r \)-partite graph \( K_{n_1, \ldots, n_r} \), then \( G \) is \( k \)-plex perfect.

**Proof.** The proof will show that all maximal \( k \)-plexes in \( G \) have the same cardinality. The result then follows from Theorem 1. Let \( K \) be a maximal \( k \)-plex in \( G \) and \( S_i \) the \( i \)th partition class. Clearly, \( |K \cap S_i| \leq |S_i| = n_i \). In addition, \( |K \cap S_i| \leq k \). For if not, let \( v \in K \cap S_i \), and
notice that \( N_G(v) \cap S_i = \emptyset \) implies

\[
\deg_{G|K}(v) = |K| - |K \cap S_i| < |K| - k,
\]

which contradicts that \( K \) is a \( k \)-plex. Therefore, \( |K \cap S_i| \leq \min\{k, n_i\} \) for each \( S_i \).

Suppose for contradiction that \( |K| = \sum_{i=1}^r |K \cap S_i| < \sum_{i=1}^r \min\{k, n_i\} \). Then there exists a \( j \) such that \( |K \cap S_j| < \min\{k, n_j\} \), and \( |K \cap S_j| < n_j \) implies that there exists a vertex \( v \in S_j \setminus K \). Consider the set \( K' := K \cup \{v\} \) and a vertex \( u \in K' \setminus S_j \). Since \( uv \in E \),

\[
\deg_{G|K'}(u) = \deg_{G|K}(u) + 1 \geq (|K| - k) + 1 = |K'| - k.
\]

Now suppose \( u \in K \cap S_j \). Observe that \( \deg_{G|K'}(u) = \deg_{G|K}(u) = |K| - |K \cap S_j| > |K| - k \) since \( uv \notin E \) and \( |K \cap S_j| < k \). It follows that

\[
\deg_{G|K'}(u) \geq |K| - k + 1 = |K'| - k.
\]

Thus, since \( \deg_{G|K'}(u) = \deg_{G|K}(v) \), \( K' \) is a \( k \)-plex in \( G \), which contradicts the maximality of \( K \). It follows that all maximal \( k \)-plexes in \( G \) have cardinality \( \sum_{i=1}^r \min\{k, n_i\} \), so \( G \) is \( k \)-plex perfect by Theorem 1. \( \square \)

It turns out that many properties of perfect graphs do not have \( k \)-plex analogues. Consider the complement \( \overline{K}_{r,r} \) of a complete bipartite graph. Both components \( H_1 \) and \( H_2 \) of \( \overline{K}_{r,r} \) are complete subgraphs.

**Lemma 2.** Let \( k \geq 1 \). If \( r = 2k - 1 \), then \( \alpha_k(\overline{K}_{r,r}) = 2k \) and \( \omega_k(\overline{K}_{r,r}) = 2k - 1 \).

**Proof.** In the proof of Theorem 2, it was shown that

\[
\omega_k(K_{r,r}) = \sum_{i=1}^2 \min\{k, r\} = 2k.
\]

Thus, \( \alpha_k(\overline{K}_{r,r}) = \omega_k(K_{r,r}) = 2k \).
Now $\omega_k(\overline{K}_{r,r}) \geq 2k - 1$ because each component $H_i$ is complete and hence a $k$-plex of cardinality $2k - 1$. Suppose for contradiction that $\omega_k(\overline{K}_{r,r}) > 2k - 1$. Then there exists a $k$-plex $K \subseteq V$ such that $|K| = 2k$. If $|K \cap H_i| \leq k$, then

$$\deg_{\overline{K}_{r,r}[K]}(v) \leq k - 1 < k = |K| - k \text{ for all } v \in K \cap H_i.$$ 

This contradicts the definition of $k$-plex. Therefore, $|K \cap H_1| > k$ and $|K \cap H_2| > k$, which contradicts $|K| = 2k$. \qed

**Theorem 3.** Let $k > 1$. If $r = 2k - 1$, then $\overline{K}_{r,r}$ is not $k$-plex perfect.

**Proof.** By Lemma 2, it suffices to show that $\chi_k(\overline{K}_{r,r}) \geq 2k$. Clearly, $\chi_k(\overline{K}_{r,r}) \geq \omega_k(\overline{K}_{r,r}) = 2k - 1$. Suppose for contradiction that $\chi_k(\overline{K}_{r,r}) = 2k - 1$. Lemma 1 implies the existence of a $\chi_k$-optimal coloring $S_1, ..., S_m$ of $\overline{K}_{r,r}$ such that $|S_1| \geq k$. Therefore, $\omega_k(\overline{K}_{r,r}[S_1]) \geq k$. Furthermore, $\chi_k(\overline{K}_{r,r}) < 2k$ implies that all other sets $S_i$ satisfy $|S_i| < k$. Notice that

$$2k - 1 = \chi_k(\overline{K}_{r,r}) = \sum_{i=1}^{m} \omega_k(\overline{K}_{r,r}[S_i]) \geq k + \sum_{i=2}^{m} \omega_k(\overline{K}_{r,r}[S_i]) = k + \sum_{i=2}^{m} |S_i|.$$ 

Consequently, $k - 1 \geq \sum_{i=2}^{m} |S_i|$. Now since the sets $S_i$ partition $V$ and $|V| = 4k - 2$,

$$|S_1| = |V| - \sum_{i=2}^{m} |S_i| \geq 3k - 1.$$ 

Therefore, $k > 1$ implies that $|S_1| \geq 3k - 1 > 2k$. This contradicts Lemma 2 because $S_1$ is a co-$k$-plex and $\alpha_k(\overline{K}_{r,r}) = 2k$. \qed

Lovász’s [16] replication lemma is a well-known result from the theory of perfect graphs. Replication of a vertex $v \in V$ corresponds to the following operation: create a new vertex $v'$ and join it to $v$ and all the neighbors of $v$. The replication lemma states that replication of a vertex in a perfect graph produces another perfect graph. However, for $k \geq 2$, replication of a vertex in a $k$-plex perfect graph does not necessarily produce another $k$-plex perfect graph.
Fix $k > 1$. Consider the edgeless graph $G$ on two vertices $v_1$ and $v_2$. $G$ is a co-$k$-plex since $\Delta(G) = 0$. It follows that $G$ is $k$-plex perfect. Construct $G'$ by performing $2k - 2$ replication operations on each of $v_1$ and $v_2$. This procedure creates $G' = \overline{K}_{r,r}$, which is not $k$-plex perfect by Theorem 3. Therefore, vertex replication does not preserve $k$-plex perfection. Theorem 3 also illustrates the following interesting property: $G$ might not be $k$-plex perfect even if all components of $G$ are $k$-plex perfect. This statement follows from Corollary 1 and Theorem 3.

The final topic of this section is a $k$-plex version of the Weak Perfect Graph Theorem [16]. The Weak Perfect Graph Theorem states that $G$ is perfect if and only if $\overline{G}$ is perfect. Theorems 2 and 3 together provide counterexamples for $k$-plex analogues of the Weak Perfect Graph Theorem for any $k \geq 2$.

### 3.2 A theorem of Erdős

In 1959, Erdős [10] showed that the difference $\chi_1(G) - \omega_1(G)$ can be arbitrarily large. More precisely, he showed that for every integer $r \geq 1$, there exists a graph $G'$ with girth $g(G') > r$ and chromatic number $\chi(G') > r$. Observe that $g(G) > 3$ implies $\omega_1(G) \leq 2$. Therefore, the theorem establishes the existence of graphs with high chromatic number and low clique number.

Analogously, one might ask if the gap between $\chi_k(G)$ and $\omega_k(G)$ can also become arbitrarily large. This section uses the Erdős theorem to show that $\chi_k(G) - \omega_k(G)$ can be arbitrarily large. The proofs have been adapted from [8]. Let $G_{n,p}$ be the random graph on $n$ vertices where each edge exists with probability $0 \leq p \leq 1$. Let $q = 1 - p$.

**Lemma 3.** Every co-$k$-plex $S$ has at most $\frac{|S| (k - 1)}{2}$ edges.

**Proof.** $|E(S)| = \frac{1}{2} \cdot \sum_{v \in V(S)} deg_{G[S]}(v) \leq \frac{1}{2} \cdot \sum_{v \in V(S)} (k - 1) = \frac{|S| (k - 1)}{2}$, where the inequality follows from the definition of co-$k$-plex. \qed
Lemma 4. For all integers \( n \geq t \geq k + 1 \), the probability that \( G \in \mathcal{G}_{n,p} \) has a co-\( k \)-plex of size \( t \) is at most

\[
P[\alpha_k(G) \geq t] \leq \left( \frac{n}{t} \right)^{t(t-k)/2}.
\]

**Proof.** Consider a fixed \( t \)-set \( U \subseteq V \). By Lemma 3, the event that \( U \) is a co-\( k \)-plex is contained in the event that

\[
|E(G[U])| \leq \frac{t(k-1)}{2}.
\]

Thus, the probability of the latter is an upper bound on the probability of the former. Now (3) requires that at least \( \binom{t}{2} - \frac{t(k-1)}{2} \) edges are missing. That is, (3) occurs with probability at most \( q^{(\frac{t}{2}) - \frac{t(k-1)}{2}} \). The lemma follows from the fact that \( G \) contains \( \binom{n}{t} \) \( t \)-sets \( U \). \( \square \)

It is worth mentioning that if \( t \leq \min\{k, n\} \), then every \( t \)-set is a co-\( k \)-plex, and Lemma 4 fails.

**Theorem 4.** Given any integer \( r > k \), there exists a graph \( G \) with girth \( g(G) > r \) and co-\( k \)-plex chromatic number \( \chi_k(G) > r \).

**Proof.** For \( n \) large, suppose \( t \geq \frac{n}{2r} > k \) and \( (8r \ln n)n^{-1} \leq p \leq 1 \). By Lemma 4,

\[
P[\alpha_k \geq t] \leq \left( \frac{n}{t} \right)^{t(t-k)/2} \leq n^t q^{t(t-k)/2} = (n q^{(t-k)/2})^t \leq (n e^{-p(t-k)/2})^t.
\]

Therefore,

\[
P[\alpha_k \geq t] \leq (n e^{-p(t-k/2)^2})^t \leq (n e^{-2(ln n)e^{k/2}})^t = (n^{-1} e^{k/2})^t.
\]

Now \( \lim_{n \to \infty} n^{-1} e^{k/2} = 0 \), so

\[
P[\alpha_k \geq \frac{n}{2r}] < \frac{1}{2}
\]

for sufficiently large \( n \).

On the other hand, fix \( \epsilon \) with \( 0 < \epsilon < 1/r \), and let \( X(G) \) denote the number of cycles of length at most \( r \) in \( G \in \mathcal{G}_{n,p} \). Erdős showed (see [8]) that for large \( n \) and \( p = n^{\epsilon-1} \),
\[ P[X \geq \frac{n}{2}] < \frac{1}{2}. \quad (5) \]

Finally, fix \( n \) large enough to satisfy (4), (5), and \( n^{\epsilon-1} \geq (8r \ln n)n^{-1} \). Let \( p = n^{\epsilon-1} \). There exists a \( G \in \mathcal{G}_{n,p} \) such that \( \alpha_k(G) < \frac{n}{2r} \) and \( G \) has less than \( \frac{n}{2} \) cycles of length at most \( r \). Construct the graph \( H \) by removing a vertex from each cycle of length at most \( r \). Then \( |H| \geq \frac{n}{2} \) and \( g(H) > r \). Furthermore, \( \alpha_k(H) \leq \alpha_k(G) < \frac{n}{2r} \) implies that any co-\( k \)-plex coloring of \( H \) requires more than \( r \) co-\( k \)-plex sets. Consequently, \( \chi_k(H) > r \).

**Corollary 2.** Given any integer \( r > k + 2 \), there exists a graph \( G \) with \( \chi_k(G) > r \) and \( \omega_k(G) < k + 2 \).

**Proof.** If \( \omega_k(G) \geq k + 2 \), then \( G \) contains a \( k \)-plex \( K \) of cardinality \( k + 2 \). Moreover, \( \delta(G[K]) \geq 2 \) by definition of \( k \)-plex. It follows that \( G[K] \subseteq G \) contains a cycle of length at most \( k + 2 = |K| \). Therefore, \( g(G) > k + 2 \) implies that \( \omega_k(G) < k + 2 \). The assertion now follows from Theorem 4.

### 4 Algorithms

This section develops algorithms for finding degree-bounded vertex partitions. Section 4.1 contains an exact \( \bar{\chi}_k \)-coloring algorithm. Section 4.2 shows how to find \( \chi_2 \)-optimal colorings using the \( \bar{\chi}_2 \)-coloring algorithm. Sections 4.1 and 4.2 both contain computational results. All implementations were run on a 2.2 GHz Dual-Core AMD Opteron processor with 3 GB of memory.

#### 4.1 \( \bar{\chi}_k \)-optimal coloring

In [15], Kubale and Jackowski present a generalized implicit enumeration algorithm for graph coloring which subsumes a number of previous combinatorial approaches [3, 4, 5, 14]. This section adapts the Kubale and Jackowski algorithm to find \( \bar{\chi}_k \)-optimal colorings.
function implicitENUM($G, n$)  
1. $ub = n + 1; r = 1$
2. loop
3. FORWARDS($r$)
4. BACKWARDS($r$)
5. if $r = 0$ then break
6. repeat
eend

function FORWARDS($r$)  
7. for $i = r$ to $n$
8. reorder uncolored vertices $v_i, ..., v_n$
9. if $r = 1$ or $r < i$ then determine $FC(i)$
10. if $FC(i) = \emptyset$ then $r = i; \text{return}$
11. $C'(i) = \min(FC(i))$
12. repeat
13. $C = C'; ub = \max(C)$
14. $r =$ least $i$ such that $C(i) = ub$
eend

function BACKWARDS($r$)  
15. $CP = \{1, ..., r - 1\}$
16. while $CP \neq \emptyset$
17. $i = \max(CP); CP = CP - \{i\}$
18. $FC(i) = FC(i) - \{C'(i)\}$
19. if $FC(i) \neq \emptyset$ then $r = i; \text{return}$
20. repeat
21. $r = 0$
eend

Figure 1: A generalized implicit enumeration algorithm [15].
Figure 4.1 contains the generalized implicit enumeration algorithm as given in [15]. Before running the algorithm, the vertex set of \( G \) is ordered \((v_1, \ldots, v_n)\) such that \( \deg_G(v_i) \geq \deg_G(v_{i+1}) \). The vertex ordering can either remain static or change dynamically throughout the algorithm. The array \( C' \) stores a partial co-\( k \)-plex coloring. The array \( C \) stores the incumbent co-\( k \)-plex coloring. For each \( 1 \leq i \leq n \), \( FC(i) \) stores the set of feasible colors for \( v_i \) with respect to the current partial coloring \( C' \). In other words, \( FC(i) \) consists of partition classes \( S \) such that \( S \cup \{v_i\} \) is a co-\( k \)-plex. \( CP \) is the set of current predecessors. These vertices are the candidates for backtracking.

The main difference between traditional graph coloring and \( \bar{\chi}_k \)-coloring is the structure of the partition classes, so adapting the coloring algorithm in Figure 4.1 amounts to finding an appropriate definition for the set of feasible colors \( FC(i) \). Given the partial co-\( k \)-plex coloring \( S_1, \ldots, S_r \), define

\[
P(i) = \{ j : S_j \cup \{v_i\} \text{ is not a co-}k\text{-plex} \} \cup \{ub\}.
\]

The set of feasible colors is defined as \( FC(i) = \{1, 2, \ldots, \max_{j<i}C'(j) + 1\} - P(i) \). This definition forces each partition class to be a co-\( k \)-plex.

For \( k = 2, 3, 4 \), the algorithm was tested on a set of random graphs and a subset of the DIMACS coloring instances. The random graph \( GN-P \) has \( N \) vertices and edge probability \( \frac{p}{100} \). Table 1 contains a description of the test instances.

Tables 2 and 3 contain computational results obtained by running two versions of this algorithm. In the first version, \( A1 \), the vertex ordering remains static. For the second version, \( A2 \), the vertex ordering is dynamic. In determining the order, the algorithm always colors the vertex \( v_i \) such that \( |P(i)| \) is maximum. Ties are broken by choosing the vertex of larger degree. This dynamic reordering is analogous to DSATUR [3].

Both algorithms appear to perform better on sparse graphs. \( A2 \) dominates \( A1 \) on all instances except for the graph \( jean \). This suggests that it is worthwhile to use a dynamic


Table 1: Test instances

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$G$</td>
<td>$</td>
<td>V</td>
<td>$</td>
<td>$</td>
<td>E</td>
</tr>
<tr>
<td>G20-10</td>
<td>20</td>
<td>21</td>
<td>G80-10</td>
<td>80</td>
<td>311</td>
</tr>
<tr>
<td>G20-30</td>
<td>20</td>
<td>56</td>
<td>G80-30</td>
<td>80</td>
<td>919</td>
</tr>
<tr>
<td>G20-50</td>
<td>20</td>
<td>98</td>
<td>G80-50</td>
<td>80</td>
<td>1583</td>
</tr>
<tr>
<td>G20-70</td>
<td>20</td>
<td>127</td>
<td>G80-70</td>
<td>80</td>
<td>2191</td>
</tr>
<tr>
<td>G20-90</td>
<td>20</td>
<td>170</td>
<td>G80-90</td>
<td>80</td>
<td>2876</td>
</tr>
<tr>
<td>G40-10</td>
<td>40</td>
<td>65</td>
<td>myciel3*</td>
<td>11</td>
<td>20</td>
</tr>
<tr>
<td>G40-30</td>
<td>40</td>
<td>223</td>
<td>myciel4*</td>
<td>23</td>
<td>71</td>
</tr>
<tr>
<td>G40-50</td>
<td>40</td>
<td>379</td>
<td>myciel5*</td>
<td>47</td>
<td>236</td>
</tr>
<tr>
<td>G40-70</td>
<td>40</td>
<td>539</td>
<td>myciel6*</td>
<td>95</td>
<td>755</td>
</tr>
<tr>
<td>G40-90</td>
<td>40</td>
<td>709</td>
<td>jean*</td>
<td>80</td>
<td>508</td>
</tr>
<tr>
<td>G60-10</td>
<td>60</td>
<td>173</td>
<td>queen5-5*</td>
<td>25</td>
<td>320</td>
</tr>
<tr>
<td>G60-30</td>
<td>60</td>
<td>546</td>
<td>queen6-6*</td>
<td>36</td>
<td>580</td>
</tr>
<tr>
<td>G60-50</td>
<td>60</td>
<td>902</td>
<td>anna*</td>
<td>138</td>
<td>986</td>
</tr>
<tr>
<td>G60-70</td>
<td>60</td>
<td>1244</td>
<td>homer*</td>
<td>561</td>
<td>3258</td>
</tr>
<tr>
<td>G60-90</td>
<td>60</td>
<td>1587</td>
<td>games120*</td>
<td>120</td>
<td>1276</td>
</tr>
</tbody>
</table>

* DIMACS graph

reordering scheme.

4.2 \( \chi_2 \)-optimal coloring

Section 4.1 shows how traditional graph coloring algorithms can solve the \( \bar{\chi}_k \)-coloring problem. However, these algorithms do not apply directly to \( \chi_k \)-coloring since a \( \chi_k \)-coloring algorithm must consider the weight \( \omega_k(G[S_i]) \) of each partition class in a co-\( k \)-plex coloring \( S_1, \ldots, S_m \).

This section focuses on the \( \chi_2 \)-coloring problem. The proposed algorithm effectively reduces \( \chi_2 \)-coloring to \( \bar{\chi}_2 \)-coloring. Lemma 1 implies that the set of compact co-\( k \)-plex colorings always contains a \( \chi_k \)-optimal coloring. As a result, an algorithm can restrict the search for a \( \chi_k \)-optimal solution by considering only compact co-\( k \)-plex colorings. For \( k = 2 \), Lemma 1 has an even stronger consequence. Recall that \( \omega_2(G[C]) = \min\{2, |C|\} \in \{1, 2\} \) for any nonempty co-2-plex \( C \) [17].

Let \( \Pi_{\text{min}} \) be the set of compact \( \bar{\chi}_2 \)-optimal colorings. Define \( \bar{C}_1, \ldots, \bar{C}_m \) as follows: If possible, choose a co-2-plex coloring in \( \Pi_{\text{min}} \) with a deficient set, and let \( \bar{C}_m \) be that deficient set. If no such co-2-plex coloring exists, choose an arbitrary co-2-plex coloring in \( \Pi_{\text{min}} \).

Lemma 5. \( \bar{C}_1, \ldots, \bar{C}_m \) is a \( \chi_2 \)-optimal coloring.
Table 2: $\bar{\chi}_k$-coloring Algorithm A1

<table>
<thead>
<tr>
<th>$G$</th>
<th>$\bar{\chi}_2(G)$</th>
<th>sec.</th>
<th>BBN</th>
<th>$\bar{\chi}_3(G)$</th>
<th>sec.</th>
<th>BBN</th>
<th>$\bar{\chi}_4(G)$</th>
<th>sec.</th>
<th>BBN</th>
</tr>
</thead>
<tbody>
<tr>
<td>G20-10</td>
<td>2</td>
<td>0</td>
<td>5</td>
<td>2</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>G20-30</td>
<td>3</td>
<td>0</td>
<td>15</td>
<td>3</td>
<td>0</td>
<td>22</td>
<td>2</td>
<td>0</td>
<td>13</td>
</tr>
<tr>
<td>G20-50</td>
<td>4</td>
<td>0</td>
<td>32</td>
<td>3</td>
<td>0</td>
<td>552</td>
<td>3</td>
<td>0</td>
<td>99</td>
</tr>
<tr>
<td>G20-70</td>
<td>6</td>
<td>0</td>
<td>1300</td>
<td>4</td>
<td>0</td>
<td>647</td>
<td>4</td>
<td>0</td>
<td>7339</td>
</tr>
<tr>
<td>G20-90</td>
<td>7</td>
<td>0</td>
<td>17509</td>
<td>5</td>
<td>0</td>
<td>38043</td>
<td>5</td>
<td>29</td>
<td>2765462</td>
</tr>
<tr>
<td>G40-10</td>
<td>3</td>
<td>0</td>
<td>122</td>
<td>2</td>
<td>0</td>
<td>5</td>
<td>2</td>
<td>0</td>
<td>9</td>
</tr>
<tr>
<td>G40-30</td>
<td>4</td>
<td>0</td>
<td>3338</td>
<td>4</td>
<td>0</td>
<td>18399</td>
<td>3</td>
<td>0</td>
<td>401</td>
</tr>
<tr>
<td>G40-50</td>
<td>6</td>
<td>2</td>
<td>219385</td>
<td>4</td>
<td>12</td>
<td>1030246</td>
<td>4</td>
<td>5</td>
<td>390503</td>
</tr>
<tr>
<td>G40-90</td>
<td>9* $\geq 1000$</td>
<td>7* $\geq 1000$</td>
<td>6* $\geq 1000$</td>
<td>63809697</td>
<td>5* $\geq 1000$</td>
<td>18335449</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G60-10</td>
<td>3</td>
<td>0</td>
<td>122</td>
<td>2</td>
<td>0</td>
<td>5</td>
<td>2</td>
<td>0</td>
<td>2875</td>
</tr>
<tr>
<td>G60-30</td>
<td>6* $\geq 1000$</td>
<td>5* $\geq 1000$</td>
<td>7* $\geq 1000$</td>
<td>41843769</td>
<td>4* $\geq 1000$</td>
<td>2875</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G60-50</td>
<td>9* $\geq 1000$</td>
<td>8* $\geq 1000$</td>
<td>7* $\geq 1000$</td>
<td>44186819</td>
<td>7* $\geq 1000$</td>
<td>18335449</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G60-70</td>
<td>14* $\geq 1000$</td>
<td>12* $\geq 1000$</td>
<td>11* $\geq 1000$</td>
<td>35905870</td>
<td>11* $\geq 1000$</td>
<td>2875</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G60-90</td>
<td>22* $\geq 1000$</td>
<td>17* $\geq 1000$</td>
<td>13* $\geq 1000$</td>
<td>31400847</td>
<td>13* $\geq 1000$</td>
<td>2875</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G80-10</td>
<td>3</td>
<td>80</td>
<td>4140070</td>
<td>3</td>
<td>0</td>
<td>3271</td>
<td>3</td>
<td>90</td>
<td>4184865</td>
</tr>
<tr>
<td>G80-30</td>
<td>7* $\geq 1000$</td>
<td>7* $\geq 1000$</td>
<td>5* $\geq 1000$</td>
<td>46164683</td>
<td>5* $\geq 1000$</td>
<td>2875</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G80-50</td>
<td>12* $\geq 1000$</td>
<td>11* $\geq 1000$</td>
<td>9* $\geq 1000$</td>
<td>25009677</td>
<td>9* $\geq 1000$</td>
<td>2875</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G80-70</td>
<td>17* $\geq 1000$</td>
<td>15* $\geq 1000$</td>
<td>12* $\geq 1000$</td>
<td>23996017</td>
<td>12* $\geq 1000$</td>
<td>2875</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G80-90</td>
<td>32* $\geq 1000$</td>
<td>23* $\geq 1000$</td>
<td>18* $\geq 1000$</td>
<td>23367032</td>
<td>18* $\geq 1000$</td>
<td>2875</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mycie3</td>
<td>2</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>0</td>
<td>6</td>
<td>2</td>
<td>0</td>
<td>8</td>
</tr>
<tr>
<td>mycie4</td>
<td>3</td>
<td>0</td>
<td>32</td>
<td>2</td>
<td>0</td>
<td>7</td>
<td>2</td>
<td>0</td>
<td>8</td>
</tr>
<tr>
<td>mycie5</td>
<td>4</td>
<td>0</td>
<td>24409</td>
<td>3</td>
<td>0</td>
<td>608</td>
<td>3</td>
<td>0</td>
<td>4888</td>
</tr>
<tr>
<td>mycie6</td>
<td>6* $\geq 1000$</td>
<td>4* $\geq 1000$</td>
<td>4* $\geq 1000$</td>
<td>13670377</td>
<td>4* $\geq 1000$</td>
<td>2875</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>jean</td>
<td>5</td>
<td>4</td>
<td>488763</td>
<td>4</td>
<td>18</td>
<td>1803128</td>
<td>3</td>
<td>9</td>
<td>803339</td>
</tr>
<tr>
<td>anna</td>
<td>6</td>
<td>0</td>
<td>8525</td>
<td>4</td>
<td>8</td>
<td>320375</td>
<td>4</td>
<td>0</td>
<td>7613</td>
</tr>
<tr>
<td>queen5-5</td>
<td>5</td>
<td>1</td>
<td>56195</td>
<td>4</td>
<td>0</td>
<td>3393</td>
<td>3</td>
<td>0</td>
<td>4841</td>
</tr>
<tr>
<td>queen6-6</td>
<td>6</td>
<td>2</td>
<td>159863</td>
<td>4</td>
<td>37</td>
<td>3099994</td>
<td>4</td>
<td>3</td>
<td>170639</td>
</tr>
<tr>
<td>homer</td>
<td>9* $\geq 1000$</td>
<td>6* $\geq 1000$</td>
<td>5* $\geq 1000$</td>
<td>1199265</td>
<td>5* $\geq 1000$</td>
<td>2875</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>games120</td>
<td>5* $\geq 1000$</td>
<td>4* $\geq 1000$</td>
<td>4* $\geq 1000$</td>
<td>10810550</td>
<td>4* $\geq 1000$</td>
<td>2875</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* upper bound
### Table 3: $\bar{\chi}_k$-coloring Algorithm $A2$

<table>
<thead>
<tr>
<th>$G$</th>
<th>$\bar{\chi}_2(G)$</th>
<th>sec.</th>
<th>BBN</th>
<th>$\bar{\chi}_3(G)$</th>
<th>sec.</th>
<th>BBN</th>
<th>$\bar{\chi}_4(G)$</th>
<th>sec.</th>
<th>BBN</th>
</tr>
</thead>
<tbody>
<tr>
<td>G20-10</td>
<td>2</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>G20-30</td>
<td>3</td>
<td>0</td>
<td>4</td>
<td>3</td>
<td>0</td>
<td>15</td>
<td>2</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>G20-50</td>
<td>4</td>
<td>0</td>
<td>16</td>
<td>3</td>
<td>0</td>
<td>224</td>
<td>3</td>
<td>0</td>
<td>40</td>
</tr>
<tr>
<td>G20-70</td>
<td>6</td>
<td>0</td>
<td>952</td>
<td>4</td>
<td>0</td>
<td>499</td>
<td>4</td>
<td>0</td>
<td>5910</td>
</tr>
<tr>
<td>G20-90</td>
<td>7</td>
<td>0</td>
<td>15575</td>
<td>5</td>
<td>1</td>
<td>30893</td>
<td>5</td>
<td>66</td>
<td>2628085</td>
</tr>
<tr>
<td>G40-10</td>
<td>3</td>
<td>0</td>
<td>7</td>
<td>2</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>G40-30</td>
<td>4</td>
<td>0</td>
<td>45</td>
<td>4</td>
<td>0</td>
<td>687</td>
<td>3</td>
<td>0</td>
<td>51</td>
</tr>
<tr>
<td>G40-50</td>
<td>6</td>
<td>0</td>
<td>1688</td>
<td>5</td>
<td>4</td>
<td>36617</td>
<td>4</td>
<td>2</td>
<td>20637</td>
</tr>
<tr>
<td>G40-70</td>
<td>9</td>
<td>289</td>
<td>2850720</td>
<td>7*</td>
<td>$\geq$1000</td>
<td>9495168</td>
<td>6*</td>
<td>$\geq$1000</td>
<td>8173137</td>
</tr>
<tr>
<td>G40-90</td>
<td>14*</td>
<td>$\geq$1000</td>
<td>10867861</td>
<td>10*</td>
<td>$\geq$1000</td>
<td>11471697</td>
<td>9*</td>
<td>$\geq$1000</td>
<td>12516863</td>
</tr>
<tr>
<td>G60-10</td>
<td>3</td>
<td>0</td>
<td>4</td>
<td>3</td>
<td>0</td>
<td>37</td>
<td>2</td>
<td>0</td>
<td>9</td>
</tr>
<tr>
<td>G60-30</td>
<td>6</td>
<td>58</td>
<td>212012</td>
<td>5*</td>
<td>$\geq$1000</td>
<td>3885427</td>
<td>4</td>
<td>98</td>
<td>366802</td>
</tr>
<tr>
<td>G60-50</td>
<td>9*</td>
<td>$\geq$1000</td>
<td>3660033</td>
<td>8*</td>
<td>$\geq$1000</td>
<td>3587213</td>
<td>7*</td>
<td>$\geq$1000</td>
<td>3902276</td>
</tr>
<tr>
<td>G60-70</td>
<td>13*</td>
<td>$\geq$1000</td>
<td>3922763</td>
<td>11*</td>
<td>$\geq$1000</td>
<td>3951519</td>
<td>11*</td>
<td>$\geq$1000</td>
<td>4561966</td>
</tr>
<tr>
<td>G60-90</td>
<td>22*</td>
<td>$\geq$1000</td>
<td>5849580</td>
<td>17*</td>
<td>$\geq$1000</td>
<td>6241803</td>
<td>12*</td>
<td>$\geq$1000</td>
<td>5917567</td>
</tr>
<tr>
<td>G80-10</td>
<td>3</td>
<td>0</td>
<td>1078</td>
<td>3</td>
<td>0</td>
<td>63</td>
<td>3</td>
<td>3</td>
<td>6717</td>
</tr>
<tr>
<td>G80-30</td>
<td>7*</td>
<td>$\geq$1000</td>
<td>2233375</td>
<td>6*</td>
<td>$\geq$1000</td>
<td>2168861</td>
<td>5*</td>
<td>$\geq$1000</td>
<td>2354629</td>
</tr>
<tr>
<td>G80-50</td>
<td>11*</td>
<td>$\geq$1000</td>
<td>1827244</td>
<td>10*</td>
<td>$\geq$1000</td>
<td>2203187</td>
<td>8*</td>
<td>$\geq$1000</td>
<td>2081951</td>
</tr>
<tr>
<td>G80-70</td>
<td>16*</td>
<td>$\geq$1000</td>
<td>2210048</td>
<td>14*</td>
<td>$\geq$1000</td>
<td>2418094</td>
<td>13*</td>
<td>$\geq$1000</td>
<td>2841362</td>
</tr>
<tr>
<td>G80-90</td>
<td>31*</td>
<td>$\geq$1000</td>
<td>3214420</td>
<td>23*</td>
<td>$\geq$1000</td>
<td>4570884</td>
<td>18*</td>
<td>$\geq$1000</td>
<td>4500137</td>
</tr>
<tr>
<td>myciel3</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>myciel4</td>
<td>3</td>
<td>0</td>
<td>9</td>
<td>2</td>
<td>0</td>
<td>11</td>
<td>2</td>
<td>0</td>
<td>14</td>
</tr>
<tr>
<td>myciel5</td>
<td>4</td>
<td>0</td>
<td>413</td>
<td>3</td>
<td>0</td>
<td>85</td>
<td>3</td>
<td>0</td>
<td>254</td>
</tr>
<tr>
<td>myciel6</td>
<td>5*</td>
<td>$\geq$1000</td>
<td>2374317</td>
<td>4</td>
<td>921</td>
<td>1997558</td>
<td>4*</td>
<td>$\geq$1000</td>
<td>1931171</td>
</tr>
<tr>
<td>jean</td>
<td>6*</td>
<td>$\geq$1000</td>
<td>7052818</td>
<td>5*</td>
<td>$\geq$1000</td>
<td>8001468</td>
<td>4*</td>
<td>$\geq$1000</td>
<td>8976619</td>
</tr>
<tr>
<td>anna</td>
<td>6</td>
<td>0</td>
<td>1058</td>
<td>4</td>
<td>1</td>
<td>2183</td>
<td>4</td>
<td>1</td>
<td>5977</td>
</tr>
<tr>
<td>queen5-5</td>
<td>5</td>
<td>0</td>
<td>1838</td>
<td>4</td>
<td>0</td>
<td>461</td>
<td>3</td>
<td>0</td>
<td>416</td>
</tr>
<tr>
<td>queen6-6</td>
<td>6</td>
<td>1</td>
<td>8916</td>
<td>4</td>
<td>22</td>
<td>194542</td>
<td>4</td>
<td>3</td>
<td>25140</td>
</tr>
<tr>
<td>homer</td>
<td>9*</td>
<td>$\geq$1000</td>
<td>314865</td>
<td>6*</td>
<td>$\geq$1000</td>
<td>425440</td>
<td>5*</td>
<td>$\geq$1000</td>
<td>503723</td>
</tr>
<tr>
<td>games120</td>
<td>5*</td>
<td>$\geq$1000</td>
<td>3109430</td>
<td>4*</td>
<td>$\geq$1000</td>
<td>2123862</td>
<td>3</td>
<td>0</td>
<td>606</td>
</tr>
</tbody>
</table>

* upper bound
Proof. Suppose for contradiction that $\tilde{C}_1, ..., \tilde{C}_m$ is not $\chi_2$-optimal. $\tilde{C}_1, ..., \tilde{C}_m$ is compact, so $|\tilde{C}_i| \geq 2$ for $i = 1, ..., m - 1$. Therefore, $\sum_{i=1}^{m} \omega_2(G[\tilde{C}_i]) = 2(m - 1) + \min\{2, |\tilde{C}_m|\}$. Now consider a $\chi_2$-optimal coloring $C_1, ..., C_r$. By Lemma 1, assume that $C_1, ..., C_r$ is compact. Therefore, $\chi_2(G) = \sum_{i=1}^{r} \omega_2(G[C_i]) = 2(r - 1) + \min\{2, |C_r|\}$, where $C_r$ is the deficient set if one exists in $C_1, ..., C_r$. Finally, observe that

$$2(m - 1) + \min\{2, |\tilde{C}_m|\} > \chi_2(G) = 2(r - 1) + \min\{2, |C_r|\},$$

which implies

$$\min\{2, |\tilde{C}_m|\} - \min\{2, |C_r|\} > 2(r - m).$$

**Case 1:** Suppose $r = m$. It follows that $\min\{2, |\tilde{C}_m|\} - \min\{2, |C_r|\} > 0$. Thus $\min\{2, |\tilde{C}_m|\} = 2$ and $\min\{2, |C_r|\} = 1$. In other words, $\tilde{C}_m$ is not deficient; $C_r$ is deficient; and $C_1, ..., C_r$ belongs to $\Pi_{\text{min}}$. This contradicts the choice of $\tilde{C}_1, ..., \tilde{C}_m$.

**Case 2:** Suppose $r > m$. Then $\min\{2, |\tilde{C}_m|\} - \min\{2, |C_r|\} > 2$, a contradiction since $\min\{2, |\tilde{C}_m|\} \in \{1, 2\}$ and $\min\{2, |C_r|\} \in \{1, 2\}$.

Lemma 5 reduces the problem of finding a $\chi_2$-optimal coloring to that of finding an element in $\Pi_{\text{min}}$. This is desirable as the algorithm from Section 4.1 can solve the latter problem. The proposed algorithm consists of two steps, both of which make use of implicit enumeration.

The first step constructs a compact $\tilde{\chi}_2$-optimal coloring. The second step searches for a compact $\tilde{\chi}_2$-optimal coloring with one deficient set. If such a co-2-plex coloring exists, it is $\chi_2$-optimal by Lemma 5. If no such co-2-plex coloring exists, then the co-2-plex coloring from the first step is $\chi_2$-optimal by Lemma 5.

The first step uses the coloring algorithm exactly as described in Section 4.1. In the second step, for each $v \in V$, the algorithm uses implicit enumeration to search for a $\tilde{\chi}_2(G) - 1$ coloring of $G - v$. If step two manages to find such a coloring in $G - v$, then $v$ is a deficient
co-2-plex in a $\tilde{\chi}_2$-optimal coloring of $G$. $A2$ was used for the implicit enumeration. Table 4 contains computational results obtained by running the algorithm on the instances which were solved to $\tilde{\chi}_2$-optimality by $A2$.

Notice that most of these graphs satisfy $\chi_2(G) = 2 \cdot \tilde{\chi}_2(G)$. These are exactly the graphs where step two of the algorithm failed to find a coloring with a deficient co-2-plex. However, the algorithm did find such a coloring for the graph $G40-10$.

### 5 Conclusion

This paper studies degree-bounded vertex partitions. Section 3 studies $\chi_k$-optimal colorings, defines $k$-plex perfection, and offers examples of $k$-plex perfect graphs. It is also shown that many properties of graph perfection do not have $k$-plex analogues. Section 3.2 uses a theorem of Erdős to show that $\chi_k(G) - \omega_k(G)$ can become arbitrarily large.

This paper also derives algorithms for constructing degree-bounded vertex partitions. Section 4.1 offers a straightforward generalization of a traditional graph coloring algorithm. The resulting algorithm partitions the vertex set into the minimum number of co-$k$-plexes. Section 4.2 shows how to find $\chi_2$-optimal colorings by reducing the problem to that of finding $\tilde{\chi}_2$-optimal colorings. Lemma 5 represents a key step in the reduction. An interesting open problem is to determine if $\tilde{\chi}_k$-coloring can be used to solve $\chi_k$-coloring problems for $k > 2$.

Another avenue for future research is a polyhedral analysis of the co-$k$-plex coloring polytope.

---

**Table 4: $\chi_2$-coloring Algorithm**

<table>
<thead>
<tr>
<th>$G$</th>
<th>$\chi_2(G)$</th>
<th>sec.</th>
<th>BBN</th>
<th>$G$</th>
<th>$\chi_2(G)$</th>
<th>sec.</th>
<th>BBN</th>
</tr>
</thead>
<tbody>
<tr>
<td>G20-10</td>
<td>4</td>
<td>0</td>
<td>23</td>
<td>G60-10</td>
<td>6</td>
<td>0</td>
<td>64</td>
</tr>
<tr>
<td>G20-30</td>
<td>6</td>
<td>0</td>
<td>24</td>
<td>G60-30</td>
<td>12</td>
<td>369</td>
<td>227480</td>
</tr>
<tr>
<td>G20-50</td>
<td>8</td>
<td>0</td>
<td>76</td>
<td>G80-10</td>
<td>6</td>
<td>4</td>
<td>1158</td>
</tr>
<tr>
<td>G20-70</td>
<td>12</td>
<td>1</td>
<td>3052</td>
<td>myciel3</td>
<td>4</td>
<td>0</td>
<td>13</td>
</tr>
<tr>
<td>G20-90</td>
<td>14</td>
<td>6</td>
<td>34475</td>
<td>myciel4</td>
<td>6</td>
<td>0</td>
<td>32</td>
</tr>
<tr>
<td>G40-10</td>
<td>5</td>
<td>0</td>
<td>7</td>
<td>myciel5</td>
<td>8</td>
<td>1</td>
<td>554</td>
</tr>
<tr>
<td>G40-30</td>
<td>8</td>
<td>0</td>
<td>165</td>
<td>anna</td>
<td>12</td>
<td>23</td>
<td>15548</td>
</tr>
<tr>
<td>G40-50</td>
<td>12</td>
<td>3</td>
<td>5958</td>
<td>queen5-5</td>
<td>10</td>
<td>1</td>
<td>2217</td>
</tr>
<tr>
<td>G40-70</td>
<td>18*</td>
<td>$\geq$ 1000</td>
<td>1579353</td>
<td>queen6-6</td>
<td>12</td>
<td>9</td>
<td>14163</td>
</tr>
</tbody>
</table>

* upper bound
References


