Discussiones Mathematicae
Graph Theory xx (xxxx) 1–16
https://doi.org/10.7151/dmgt.2574

RECOGNIZABLE COLORING OF GRAPHS

Zahra Hamed-Labbafian[a], Michael A. Henning[b]

Mostafa Tavakoli[a,1] and Narjes Sabeghi[c]

[a]*Department of Applied Mathematics, Faculty of Mathematical Sciences*
Ferdowsi University of Mashhad, Mashhad, 91775 Iran

[b]*Department of Mathematics and Applied Mathematics*
University of Johannesburg
Auckland Park, 2006 South Africa

[c]*Department of Mathematics, Faculty of Basic Sciences*
Velayat University, Iranshar, Iran

e-mail: hamedlabbafianzahra@gmail.com
mahenning@uj.ac.za
m_tavakoli@um.ac.ir
sabeghinarjes@gmail.com

Abstract

Let G be a connected graph and f be a mapping from $V(G)$ to S, where S is a set of k colors for some positive integer k. The color code of a vertex v of G with respect to f, denoted by $\text{code}_G(v|f)$, is the ordered $(k+1)$-tuple $(x_0, x_1, \ldots, x_k)$ where x_0 is the color assigned to v and where x_i is the number of vertices adjacent to v of color i for $1 \leq i \leq k$, that is, $x_i = |\{uv \in E(G) : f(u) = i\}|$ for $1 \leq i \leq k$. The mapping f is a recognizable coloring if $\text{code}_G(u|f) \neq \text{code}_G(v|f)$ for every two distinct vertices u and v of G. The minimum number of colors needed for a recognizable coloring of G is the recognition number of G denoted by $\text{rn}(G)$. Our goal in this article is to give the exact value of the recognition number of the corona product $G \circ H$ of two graphs G and H for the cases $H = K_n$ and $n \geq |V(G)|$, or $G = K_m$ and $H = K_n$ with $m > n$. In addition, we obtain the exact value of the recognition number of the edge corona product $G \diamond H$ of G and H for the case that G is a non-trivial graph with minimum degree at least 2 and $H = K_n$ where $n \geq |E(G)|$. Moreover, an algorithm for computing the recognition number of graphs is presented. As an application of our algorithm, we compute the recognition number of some fullerene graphs.

Keywords: recognizable coloring, corona product, local search algorithm, fullerene.

2020 Mathematics Subject Classification: 05C15, 90C59.

---

[1]Corresponding author.

1. Introduction

All graphs considered in this paper are assumed to be simple and connected. In modeling real world problems using graph theory, we sometimes come across problems that correspond to distinguishing vertices of graphs. Harary and Plantholt [3] introduced the *point-distinguishing edge coloring* (also called a *vertex-distinguishing edge coloring* in the literature) of graph which is defined as assigning colors to the edges of the graph so that the sets of colors used on the edges incident on two different vertices are unequal.

In addition to vertex-distinguishing edge colorings, vertex-distinguishing vertex colorings have been studied. For this purpose, for a graph G and a mapping $f\colon V(G) \to \{1,\dots,k\}$ where k is a positive integer, the *color code* of a vertex v of G with respect to f, denoted by $\mathrm{code}_G(v|f)$, is the ordered $(k+1)$-tuple $(x_0, x_1, \dots, x_k)$ where x_0 is the color assigned to v and where x_i is the number of vertices adjacent to v of color i for $1 \le i \le k$, that is, $x_i = |\{uv \in E(G) : f(u) = i\}|$ for $1 \le i \le k$. A coloring f of the vertices of G is an *irregular coloring* if distinct vertices of G have distinct color codes. Formally, an irregular coloring f of G assigns colors to the vertices of G in such a way that (i) $f(u) \neq f(v)$ for each $uv \in E(G)$ and (ii) distinct vertices have distinct color codes, see [7].

Every irregular coloring of a graph G is a proper coloring of G. If we remove this requirement from an irregular coloring, that is, if we remove condition (i) from the definition of an irregular coloring, then the resulting coloring is a *recognizable coloring* of G. The minimum number of colors in a recognizable coloring on G is the *recognition number* of G denoted by $\mathrm{rn}(G)$. The concept of recognizable colorings was first introduced and studied in 2008 by Chartrand *et al.* in [1]. They proved that for each pair k, n of integers with $2 \le k \le n$, there exists a connected graph of order n with recognition number k. They also established characterizations of connected graphs of order n with recognition numbers n or $n-1$. Moreover, they presented a conjecture about the recognition number of cycles and trees, which was proved in 2012 by Dorfling and Dorfling in [2].

In the current work, for two non-trivial graphs G and H, we study the recognition number $\mathrm{rn}(G \circ H)$ in the corona product $G \circ H$ and we study $\mathrm{rn}(G \diamond H)$ in the edge corona product $G \diamond H$. We give the exact value of $\mathrm{rn}(G \circ H)$ for the cases $H = K_n$ where $n \ge |V(G)|$, and $G = K_m$ and $H = K_n$ where $m > n$. In addition, we obtain the exact value of $\mathrm{rn}(G \diamond K_n)$ for the case that the G has minimum degree at least 2. Moreover, we present an algorithm for computing the recognition number of graphs in general. As an application of our algorithm, we compute the recognition number of some fullerene graphs.

1.1. Notation

For graph theory notation and terminology, we generally follow [4]. Specifically, let G be a graph with vertex set $V(G)$ and edge set $E(G)$, and of order $|V(G)|$ and size $|E(G)|$. Two adjacent vertices in G are *neighbors*. The *open neighborhood* of a vertex v in G is $N_G(v) = \{u \in V : uv \in E\}$ and the *closed neighborhood of* v is $N_G[v] = \{v\} \cup N_G(v)$. We denote the degree of v in G by $\deg_G(v)$, and so $\deg_G(v) = |N_G(v)|$. The minimum degree among the vertices of G is denoted by $\delta(G)$. For $k \geq 1$ an integer, we let $[k]$ denote the set $\{1, \ldots, k\}$.

2. Recognizable Coloring of Corona Products

Let G and H be two graphs with $V(G) = \{g_1, \ldots, g_m\}$, and let $H_1, \ldots, H_m$ be m vertex disjoint copies of H. The *corona product* $G \circ H$ of G and H is obtained by taking one copy of G and the m copies of H and joining by an edge the vertex g_i of G to every vertex in the ith copy H_i of H for $i \in [m]$, see [6] for more details.

Lemma 2.1. *Let G be a graph of order m and let $H = K_n$. If f is a recognizable coloring of $G \circ H$, then $f(u) \neq f(v)$ for every two distinct vertices u and v in the ith copy of H for all $i \in [m]$.*

Proof. Let f be a recognizable coloring on $G \circ H$ where G has order m and $H = K_n$. Suppose, to the contrary, that there exist two distinct vertices u and v in the ith copy of H such that $f(u) = f(v)$ for some $i \in [m]$. Since H is a complete graph, we note that $N_{G\circ H}[u] = N_{G\circ H}[v]$. Thus since $f(u) = f(v)$, we infer that $\mathrm{code}_{G\circ H}(u|f) = \mathrm{code}_{G\circ H}(v|f)$, contradicting the fact that f is a recognizable coloring on $G \circ H$. ∎

Theorem 2.2. *Let G and H be two graphs of order m and n, respectively. If $1 < m \leq n$, then $\mathrm{rn}(G \circ H) \leq n$.*

Proof. To show that $\mathrm{rn}(G \circ H) \leq n$, we present a recognizable coloring f on $G \circ H$ using n colors. Let f be the coloring that assigns color i to vertex g_i for all $i \in [m]$ and assigns to the n vertices from each copy of H a color from the set $[n]$ in such a way that the vertices from each copy of H receive different colors. We prove that the resulting coloring f is a recognizable coloring of $G \circ H$. Consider two distinct vertices u and v of $G \circ H$. Note that if u and v are both in the copy of G, then we have nothing to prove because $f(u) \neq f(v)$. If u is in the copy of G and v is in a copy of H, then $\deg_{G\circ H}(u) > \deg_{G\circ H}(v)$, and so $\mathrm{code}_{G\circ H}(u|f) \neq \mathrm{code}_{G\circ H}(v|f)$. If u and v are in the same copy of H, then since $f(u) \neq f(v)$, once again we infer that $\mathrm{code}_{G\circ H}(u|f) \neq \mathrm{code}_{G\circ H}(v|f)$. Hence we may assume that u and v are in the ith and jth copies of H, respectively, where $1 \leq i < j \leq m$. In this case, the entry x_i in the color code $\mathrm{code}_{G\circ H}(u|f)$ is greater than x_i in the color code $\mathrm{code}_{G\circ H}(v|f)$, and so $\mathrm{code}_{G\circ H}(u|f) \neq \mathrm{code}_{G\circ H}(v|f)$. Hence, f is a recognizable coloring of $G \circ H$, and so $\mathrm{rn}(G \circ H) \leq n$. ∎

By the fact that $K_1 \circ K_n$ is isomorphic to K_{n+1}, we have $\mathrm{rn}(K_1 \circ K_n) = n+1$. The next result shows the bound presented in Theorem 2.2 is sharp.

Corollary 2.3. *If G is a graph of order m, then $\mathrm{rn}(G \circ K_n) = n$ where $1 < m \leq n$.*

Proof. Let G be a graph of order m where $1 < m \leq n$. If f is a recognizable coloring of $G \circ K_n$, then by Lemma 2.1 the n vertices in every copy of K_n in $G \circ K_n$ are assigned different colors, implying that $\mathrm{rn}(G \circ K_n) \geq n$. On the other hand, by Theorem 2.2 $\mathrm{rn}(G \circ K_n) \leq n$ which completes the proof. ■

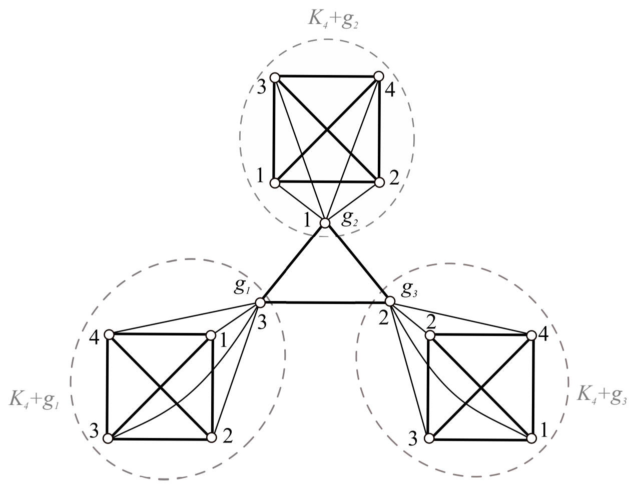

Figure 1. A recognizable coloring of $K_3 \circ K_4$ using four colors.

By Corollary 2.3, $\mathrm{rn}(K_3 \circ K_4) = 4$. A recognizable coloring of $K_3 \circ K_4$ using four colors is depicted in Figure 1.

Theorem 2.4. $\mathrm{rn}(K_m \circ K_n) = \min\{t : m \leq \binom{t}{n+1} + n\binom{t}{n}\}$ *where $m > n$.*

Proof. Let $G = K_m$ and let $H = K_n$ where $m > n$. Let $H_1, \ldots, H_m$ be the m vertex disjoint copies of H in $G \circ H$ where the ith copy H_i is associated with the ith vertex g_i of G for $i \in [m]$, that is, the vertex g_i of G is adjacent in $G \circ H$ to every vertex in the ith copy H_i of H for $i \in [m]$. Set

$$k = \min\left\{t : m \leq \binom{t}{n+1} + n\binom{t}{n}\right\}.$$

Based on Lemma 2.1, each recognizable coloring of $G \circ H$ satisfies at least one of the following cases for each $i \in [m]$.

Case 1. The vertices of $V(H_i) \cup \{g_i\}$ are colored with $n+1$ different colors.

Case 2. The vertices of $V(H_i) \cup \{g_i\}$ are colored with n different colors. (In this case, the color of vertex g_i is the same as the color of one of the vertices of $V(H_i)$.)

In the first case, there are $\binom{k}{n+1}$ ways, $\left\{c_1, \ldots, c_{\binom{k}{n+1}}\right\}$, of choosing $n+1$ colors from k possible colors for coloring the vertices of $V(H_i) \cup \{g_i\}$. As an illustration, consider the graph $K_6 \circ K_4$ shown in Figure 2. In this example, $k = 5$ and so $\left\{c_1, \ldots, c_{\binom{5}{4+1}}\right\} = \{c_1\} = \{\{1, 2, 3, 4, 5\}\}$. As depicted in this figure, the vertices from $V(H_1) \cup \{g_1\}$ are colored with the colors of $c_1 = \{\{1, 2, 3, 4, 5\}\}$.

In the second case, there are $\binom{k}{n}$ ways of choosing n colors from k possible colors for coloring the vertices of $V(H_i)$. Moreover there are $\binom{n}{1}$ ways to select a color from the n selected colors of vertices in $V(H_i)$ to color the vertex g_i. Therefore, there are $n\binom{k}{n}$ possible choices $\left\{c_{\binom{k}{n+1}+1}, \ldots, c_{\binom{k}{n+1}+n\binom{k}{n}}\right\}$ for coloring vertices of $V(H_i) \cup \{g_i\}$ in the second case. As an illustration, consider the graph $K_6 \circ K_4$ shown in Figure 2 with $k = 5$. In this example, we have

$$\begin{aligned}\{c_2, \ldots, c_{21}\} = \Big\{&\{\{1,1,2,3,4\}\}, \{\{2,1,2,3,4\}\}, \{\{3,1,2,3,4\}\}, \{\{4,1,2,3,4\}\},\\ &\{\{1,1,2,3,5\}\}, \{\{2,1,2,3,5\}\}, \{\{3,1,2,3,5\}\}, \{\{5,1,2,3,5\}\},\\ &\{\{1,1,2,4,5\}\}, \{\{2,1,2,4,5\}\}, \{\{4,1,2,4,5\}\}, \{\{5,1,2,4,5\}\},\\ &\{\{1,1,3,4,5\}\}, \{\{3,1,3,4,5\}\}, \{\{4,1,3,4,5\}\}, \{\{5,1,3,4,5\}\},\\ &\{\{2,2,3,4,5\}\}, \{\{3,2,3,4,5\}\}, \{\{4,2,3,4,5\}\}, \{\{5,2,3,4,5\}\}\Big\}.\end{aligned}$$

As shown in this figure, in the example in Figure 2 the vertices of $V(H_2) \cup \{g_2\}$, $V(H_3) \cup \{g_3\}$, $V(H_4) \cup \{g_4\}$, and $V(H_5) \cup \{g_5\}$ are colored by the colors of $c_2 = \{\{1,1,2,3,4\}\}$, $c_3 = \{\{2,1,2,3,4\}\}$, $c_4 = \{\{3,1,2,3,4\}\}$, $c_5 = \{\{4,1,2,3,4\}\}$, and $c_6 = \{\{1,1,2,3,5\}\}$, respectively.

Therefore, m must be less than or equal to $\binom{t}{n+1} + n\binom{t}{n}$ which concludes that $m \le k$ and consequently $\operatorname{rn}(K_m \circ K_n) \ge k = \min\{t : m \le \binom{t}{n+1} + n\binom{t}{n}\}$.

In the rest of the proof, we show that $\operatorname{rn}(K_m \circ K_n) \le \min\{t : m \le \binom{t}{n+1} + n\binom{t}{n}\}$. To do this, we will prove that an one-to-one mapping f from $\{V(H_1) \cup \{g_1\}, \ldots, V(H_m) \cup \{g_m\}\}$ to $\left\{c_1, \ldots, c_{\binom{k}{n+1}+n\binom{k}{n}}\right\}$ is a recognizable coloring of $G \circ H$. Let u and v be two vertices of $G \circ H$. If $f(u) \ne f(v)$, then it is immediate that $\operatorname{code}_{G\circ H}(u|f) \ne \operatorname{code}_{G\circ H}(v|f)$. Hence we may assume that $f(u) = f(v)$. We note that in our method f plays the role of a one-to-one mapping from $\{V(H_1) \cup \{g_1\}, \ldots, V(H_m) \cup \{g_m\}\}$ to $\left\{c_1, \ldots, c_{\binom{k}{n+1}+n\binom{k}{n}}\right\}$. Let G' be the copy of G in $G \circ H$.

Suppose that $u \in V(G')$ and $v \in V(H_i)$ for some $i \in [m]$. In this case, $\deg_{G\circ H}(u) = n + m - 1$ and $\deg_{G\circ H}(n) = n$. Hence since $m > n$, we infer that $\operatorname{code}_{G\circ H}(u|f) \ne \operatorname{code}_{G\circ H}(v|f)$.

Suppose that $u \in V(G')$ and $v \in V(G')$. Thus, $u = g_i$ and $v = g_j$ for some i and j where $1 \leq i < j \leq m$. In this case, $N_{G\circ H}[g_i] \cap V(G') = N_{G\circ H}[g_j] \cap V(G')$ (and recall that by assumption $f(g_i) = f(g_j)$). Since the vertices of $V(H_i) \cup \{g_i\}$ and $V(H_j) \cup \{g_j\}$ are colored by different color cases of $c_1, \ldots, c_k$, we infer that $\mathrm{code}_{G\circ H}(u|f) \neq \mathrm{code}_{G\circ H}(v|f)$.

Suppose that $u \in V(H_i)$ and $v \in V(H_j)$ for some i and j where $1 \leq i < j \leq m$. Since the vertices of $V(H_i) \cup \{g_i\}$ and $V(H_j) \cup \{g_j\}$ are colored by different color cases of $c_1, \ldots, c_k$, we once again infer that $\mathrm{code}_{G\circ H}(u|f) \neq \mathrm{code}_{G\circ H}(v|f)$.

Hence we have shown that if $f(u) = f(v)$, then in all cases $\mathrm{code}_{G\circ H}(u|f) \neq \mathrm{code}_{G\circ H}(v|f)$. Therefore, f is a recognizable coloring of $K_m \circ K_n$, which completes the proof. ∎

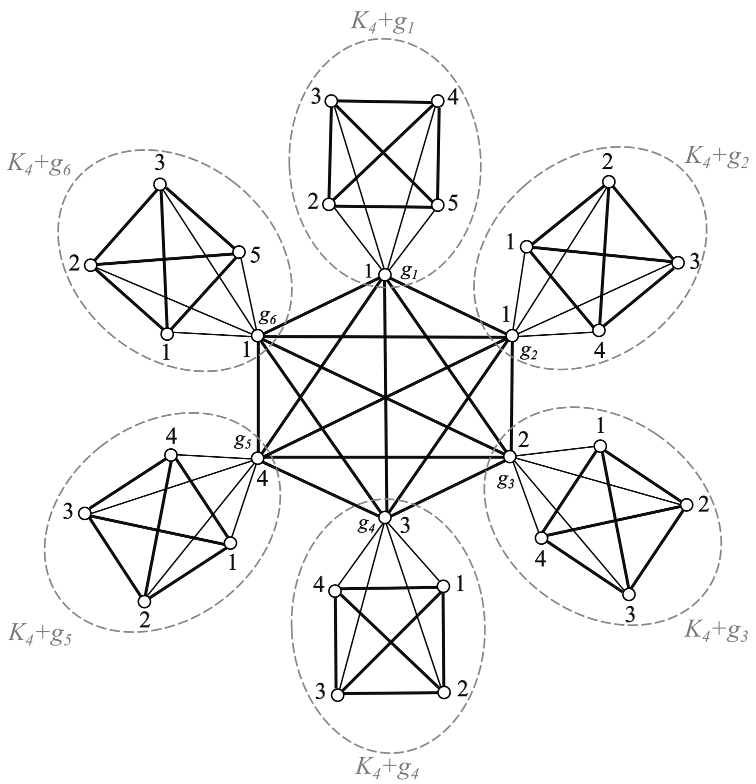

Figure 2. A recognizable coloring of $K_6 \circ K_4$.

Theorem 2.5. *Let H be a graph and $\rho = \max\left\{m, d_1^H, \ldots, d_{\Delta_H}^H\right\}$ where $m > 1$ and $d_i^H = |\{u \in V(H)\colon \deg_H(u) = i\}|$ for $i \in [\Delta_H]$. Then $\mathrm{rn}(K_m \circ H) \leq \rho$.*

Proof. Let H be a graph and $V(K_m) = \{g_1, \ldots, g_m\}$. Set $V_i = \{u \in V(H) : \deg_H(u) = i\}$ for $i \in [\Delta_H]$. Suppose that $V_i = \left\{h_{i_1}, \ldots, h_{i_{d_i}}\right\}$ for $i \in [\Delta_H]$. We define f as

$$f(u) = \begin{cases} i, & \text{if}\, u = g_i, \\ j, & \text{if}\, u\, \text{corresponds to}\, h_{i_j}. \end{cases}$$

Now we show that f is a recognizable coloring of $K_m \circ H$. Consider two vertices u and v in $K_m \circ H$. Note that we have nothing to prove for cases that u and v are both in G or in the same copy of H because $f(u) \neq f(v)$ in these cases. In addition, since $\deg_{K_m \circ H}(u) > \deg_{K_m \circ H}(v)$ where u is in K_n and v is in a copy of H we conclude that $\text{code}_{K_m \circ H}(u|f) \neq \text{code}_{K_m \circ H}(v|f)$ for this case. Hence we may assume that u and v are in the ith and tth copies of H, respectively, where $1 \leq i < t \leq m$. If $\deg_{K_m \circ H}(u) \neq \deg_{K_m \circ H}(v)$, clearly $\text{code}_{G \circ H}(u|f) \neq \text{code}_{G \circ H}(v|f)$. Hence, suppose that $\deg_{K_m \circ H}(u) = \deg_{K_m \circ H}(v)$. In this case, the entry x_i in the color code $\text{code}_{G \circ H}(u|f)$ is greater than x_i in the color code $\text{code}_{G \circ H}(v|f)$, and so $\text{code}_{G \circ H}(u|f) \neq \text{code}_{G \circ H}(v|f)$. Hence, f is a recognizable coloring of $G \circ H$, and so $\text{rn}(G \circ H) \leq \rho$. ■

3. RECOGNIZABLE COLORING OF EDGE CORONA PRODUCTS

Let G be a graph of size m. Let G and H be two graphs with $E(G) = \{e_1, \ldots, e_m\}$, and let $H_1, \ldots, H_m$ be m vertex disjoint copies of H. The *edge corona product* $G \diamond H$ of G and H is obtained from one copy of G and m copies of H as follows: for each edge $e_i = g_i g_i'$ of G we join g_i and g_i' to every vertex in the ith copy H_i of H associated with the edge e_i for $i \in [m]$. The ith copy H_i of H corresponding to the edge e_i of G we also denote by H_{e_i} for $i \in [m]$, see [6, 5] for more details.

Lemma 3.1. *Let G be a graph of size m and let $H = K_n$. If f is a recognizable coloring of $G \diamond H$, then $f(u) \neq f(v)$ for every two distinct vertices u and v in the ith copy of H for all $i \in [m]$.*

Proof. Let f be a recognizable coloring on $G \diamond H$ where G has size m and $H = K_n$. Suppose, to the contrary, that there exist two distinct vertices u and v in the ith copy of H such that $f(u) = f(v)$ for some $i \in [m]$. Since H is a complete graph, we note that $N_{G \diamond H}[u] = N_{G \diamond H}[v]$. Thus since $f(u) = f(v)$, we infer that $\text{code}_{G \diamond H}(u|f) = \text{code}_{G \diamond H}(v|f)$, contradicting the fact that f is a recognizable coloring on $G \diamond H$. ■

Theorem 3.2. *If G is a connected graph of order p and size m where $m \geq p$, then $\text{rn}(G \diamond K_n) = n$ where $n \geq m$.*

Proof. Let G be a connected graph of order p and size m where $m \geq p$ and let $H = K_n$ where $n \geq m$. We consider the edge corona product $G \diamond H$ of G and H. Let $V(G) = \{g_1, g_2, \ldots, g_p\}$ and $E(G) = \{e_1, \ldots, e_m\}$, and so $p = |V(G)|$

and $m = |E(G)|$. By supposition, $m \geq p$. Let $V(H) = \{h_1, h_2, \ldots, h_n\}$. We adopt our notation defined immediately before the statement of Lemma 3.1. In particular, $H_1, \ldots, H_m$ denote the vertex disjoint copies of H where the ith copy H_i, also denoted by H_{e_i}, of H is associated with the edge e_i for $i \in [m]$. We let $V(H_i) = \{h_{1_i}, h_{2_i}, \ldots, h_{n_i}\}$ where the vertex h_{j_i} in H_i corresponds to the vertex h_j in H for all $j \in [n]$. By Lemma 3.1, we infer that $\mathrm{rn}(G \diamond H) \geq n$.

To show that $\mathrm{rn}(G \diamond H) \leq n$, we present a recognizable coloring f on $G \diamond H$ using n colors. Since $\delta(G) \geq 2$, we note that $m = |E(G)| \geq |V(G)|$. Thus, $p = |V(G)| \leq |E(G)| = m \leq n = |V(H)|$. Let $f\colon V(G \diamond H) \to [n]$ be the coloring that assigns color i to vertex g_i for all $i \in [p]$. We note that since n colors are used and since $n \geq p$, we can indeed color the vertices of G with distinct colors. Next we color the vertex h_{j_i} that belongs to the ith copy H_i of H with the color j for all i and j where $i \in [m]$ and $j \in [n]$. We note that the n vertices in each copy H_i of H in $G \diamond H$ each receive a distinct color from the available n colors.

Let u and v be two vertices of $G \diamond H$. If $f(u) \neq f(v)$, then it is immediate that $\mathrm{code}_{G\diamond H}(u|f) \neq \mathrm{code}_{G\diamond H}(v|f)$. If one of u and v belongs to G and the other to a copy of H in $G \diamond H$, then once again we immediate have $\mathrm{code}_{G\diamond H}(u|f) \neq \mathrm{code}_{G\diamond H}(v|f)$. Hence we may assume that $f(u) = f(v)$ and that u and v belong to the different copies of H in $G \diamond H$. Suppose that u belongs to H_i and v belongs to H_r in $G \diamond H$ for some i and r where $1 \leq i < r \leq m$. Since $f(u) = f(v)$, we infer that $u = h_{j_i}$ and $v = h_{r_i}$ for some $j \in [n]$. Thus, u is the vertex in the ith copy of H corresponding to the vertex h_j of H and v is the vertex in the rth copy of H corresponding to the vertex h_j of H.

Recall that H_i is the copy of H associated with the ith edge e_i of G and H_r is the copy of H associated with the rth edge e_r of G. Let $e_i = g_i g_i'$ and let $e_r = g_r g_r'$. Since $\{g_i, g_i'\} \neq \{g_r, g_r'\}$, we may assume renaming the ends of the edges e_i and e_r if necessary, that $g_i \neq g_r$. Recall that the vertex g_i is colored with color i and the vertex g_r with color r. Thus, x_i is 2 in the color code $\mathrm{code}_{G\diamond H}(u|f)$ while x_i is 1 in the color code $\mathrm{code}_{G\diamond H}(v|f)$, and x_r is 1 in the color code $\mathrm{code}_{G\diamond H}(u|f)$ while x_r is 2 in the color code $\mathrm{code}_{G\diamond H}(v|f)$. Thus, $\mathrm{code}_{G\diamond H}(u|f) \neq \mathrm{code}_{G\diamond H}(v|f)$. Hence we have shown that if $f(u) = f(v)$, then in all cases $\mathrm{code}_{G\diamond H}(u|f) \neq \mathrm{code}_{G\diamond H}(v|f)$. Therefore, f is a recognizable coloring of $G \diamond H$, which completes the proof. ■

Using a similar technique applied in the proof of Theorem 2.4 we obtain the next result about $\mathrm{rn}(G \diamond H)$.

Proposition 3.3. $\mathrm{rn}(G \diamond K_n) \geq \min\left\{t : m \leq \binom{t}{n+2} + (n+1)\binom{t}{n+1} + \frac{n(n+1)}{2}\binom{t}{n}\right\}$.

Proof. Let $H = K_n$ and $|E(G)| = m$. Let $H_1, \ldots, H_m$ be the m vertex disjoint copies of H in $G \diamond H$ where the ith copy H_i is associated with the ith edge $e_i = g_i g_i'$ of G for $i \in [m]$, that is, end vertices of e_i of G is adjacent in $G \diamond H$ to every vertex in the ith copy H_i of H for $i \in [m]$. Set

$$\begin{aligned} k &= \min\left\{t : m \le \binom{t}{n+2} + \binom{t}{n+1}\binom{n+1}{1} + \binom{t}{n}\binom{n}{1} + \binom{t}{n}\binom{n}{2}\right\} \\ &= \min\left\{t : m \le \binom{t}{n+2} + (n+1)\binom{t}{n+1} + \frac{n(n+1)}{2}\binom{t}{n}\right\}. \end{aligned}$$

According to Lemma 3.1, each recognizable coloring of $G \diamond H$ satisfies at least one of the following cases for each $i \in [m]$.

Case 1. The vertices of $V(H_i) \cup \{g_i, g_i'\}$ are colored with $n+2$ different colors.

Case 2. The vertices of $V(H_i) \cup \{g_i, g_i'\}$ are colored with $n+1$ different colors. (In this case, the color of one of the vertices g_i and g_i, say g_i', is the same as the color of one of the vertices of $V(H_i) \cup \{g_i\}$.)

Case 3. The vertices of $V(H_i) \cup \{g_i, g_i'\}$ are colored with n different colors. (In this case, the color of vertices g_i and g_i' are the same as the color of one of the vertices of $V(H_i)$.)

In the first case, there are $\binom{t}{n+2}$ ways, $\left\{c_1, \ldots, c_{\binom{t}{n+2}}\right\}$, of choosing $n+2$ colors from t possible colors for coloring the vertices of $V(H_i) \cup \{g_i, g_i'\}$.

In the second case, there are $\binom{t}{n+1}$ ways of choosing $n+1$ colors from t possible colors for coloring the vertices of $V(H_i) \cup \{g_i\}$. Moreover there are $\binom{n+1}{1}$ ways to select a color from the $n+1$ selected colors of vertices in $V(H_i) \cup \{g_i\}$ to color the vertex g_i'. Therefore, there are $(n+1)\binom{t}{n+1}$ possible choices $\left\{c_{\binom{t}{n+2}+1}, \ldots, c_{\binom{t}{n+2}+(n+1)\binom{t}{n+1}}\right\}$ for coloring vertices of $V(H_i) \cup \{g_i, g_i'\}$ in the second case.

In the third case, there are $\binom{t}{n}$ ways of choosing n colors from t possible colors for coloring the vertices of $V(H_i)$. In addition, there are $\frac{1}{2}n(n+1)$ ways to select colors from the n selected colors of vertices in $V(H_i)$ to color vertices g_i and g_i'. Therefore, there are $\frac{1}{2}n(n+1)\binom{t}{n}$ possible choices

$$\left\{c_{\binom{t}{n+2}+(n+1)\binom{t}{n+1}+1}, \ldots, c_{\binom{t}{n+2}+(n+1)\binom{t}{n+1}+\frac{n(n+1)}{2}\binom{t}{n}}\right\}$$

for coloring vertices of $V(H_i) \cup \{g_i, g_i'\}$ in the third case.

Therefore, m must be less than or equal to $\binom{t}{n+2} + (n+1)\binom{t}{n+1} + \frac{n(n+1)}{2}\binom{t}{n}$. We therefore infer that $m \le k$ and consequently

$$\mathrm{rn}(G \diamond K_n) \ge \min\left\{t : m \le \binom{t}{n+2} + (n+1)\binom{t}{n+1} + \frac{n(n+1)}{2}\binom{t}{n}\right\},$$

as desired. ■

4. Finding a Recognizable Coloring of a Graph Using a Local Search Algorithm

In this section, we intend to present a local search algorithm for finding a recognizable coloring of a graph. This Local Search Algorithm (LSA) aims to find a recognizable coloring for a graph by iteratively improving an initial solution.

In Algorithm 1, $Maxit$ is the maximum number of iterations chosen as a stopping criterion. C is a data structure that includes the following two fields.

- Value ($C.value$): This field represents the current coloring of the graph. It is an array or list of integers, where each integer denotes the color assigned to a specific vertex of the graph.
- Cost ($C.cost$): This field is a numerical value that evaluates the quality of the current coloring, specifically measuring the number of conflicts in the coloring.

These two fields are considered for Sn, the neighbor solution of C and $Nsol$, the best neighbor solution, correspondingly.

After generating an arbitrary coloring and the corresponding map f, we create a coding matrix. The ith row of this matrix is $\mathrm{code}_G(v_i|f)$. The sub-algorithm 2 compares every two row of coding matrix and return the number of vertices for which $\mathrm{code}_G(u|f) = \mathrm{code}_G(v|f)$. The algorithm continues by generating neighboring solutions. If for a specific k, a neighbor solution is found such that the code for every two vertices is different, we have a recognizable coloring. Otherwise, it can try the algorithm for another value of k. In section 5, where we explain some applications of Algorithm 1, we assert the methodology for determining the parameter k for some graphs.

The details of the steps of this algorithm are as follows.

4.1. Explanation of the Main Algorithm (LSA)

Initialization

The algorithm begins by generating a random coloring for the graph vertices and stored in $C.value$. Then it calculates the initial number of conflicts ($C.cost$) using the sub-algorithm (CA). If no conflicts are detected in this initial coloring, it is immediately returned as the solution.

Local search loop

If conflicts exist, the algorithm enters a local search loop. In each iteration, a set of neighboring configurations (each differing from the current coloring in only one vertex's color) is generated. For each neighbor configuration, the algorithm evaluates the number of conflicts and updates the current solution if a neighboring solution with fewer conflicts is found.

Updating best solutions

The best neighboring solution is compared to the current coloring. If it results in fewer conflicts, it is added to a list of potential solutions (BestL). If a conflict-free neighbor is found, it is also added to this list as an optimal solution. If not, the algorithm sets the current coloring to the best neighboring solution and increments the iteration count.

Termination

The process continues until either a conflict-free coloring is found, the maximum number of iterations (Maxit) is reached, or no further improvement can be made. Finally, the algorithm returns the best solution from the list BestL, which contains the configuration with the lowest number of conflicts.

4.2. Explanation of the Conflict Calculation Algorithm (CA)

Coding Matrix

It generates a coding matrix (CM). The ith row of this matrix is $\mathrm{code}_G(v_i|f)$.

Conflict counting

The algorithm then compares each pair of rows in the matrix CM. If two rows are identical (indicating that two vertices have the same code), this is counted as a conflict.

Return

After processing all pairs, the algorithm returns the total count of conflicts for the current coloring configuration.

4.3. Performance considerations

The algorithm's effectiveness may vary based on the type of graph. For graphs with a high number of vertices and complex connectivity, it may take more iterations to find a low-conflict coloring or might not reach a conflict-free configuration within the set limit.

Algorithm 1 Local Search Algorithm to find a recognizable coloring (LSA)

Require: Graph G, the number of colors (k), the number of iterations, $Maxit$.
Ensure: a recognizable coloring for G.

1: Generate an empty list $BestL$. //List to store potential solutions.
2: Generate an arbitrary vertex coloring, $C.value$, for G.
3: Call sub-algorithm 2 for $C.value$ ($C.cost \leftarrow CA(C.value, n(G), k)$.
4: **if** $C.cost = 0$ **then**
5:     Return C. //Return conflict-free coloring.
6: **else**
7:     Let $count \leftarrow 1$. //count is the iteration counter.
8:     **while** $count <= Maxit$ or $C.cost \neq 0$ **do**
9:         Generate neighborhood set NC of $C.value$, where the value of each element in this set differs from $C.value$ in the color of only one vertex.
10:         $g_{best} \leftarrow C.cost$.
11:         $Nsol \leftarrow C$.
12:         **while** $NC \neq \emptyset$ **do**
13:             Select an element, $Sn.value$ of NC and Call CA for it ($Sn.cost \leftarrow CA(Sn.value, n(G), k)$).
14:             **if** $Sn.cost < g_{best}$ **then**
15:                 $Nsol \leftarrow Sn$ and $g_{best} \leftarrow Sn.cost$.
16:             **end if**
17:             $NC \leftarrow NC - \{Sn.value\}$.
18:         **end while**
19:         **if** $C.cost <= Nsol.cost$ **then**
20:             Add C to the $BestL$.
21:         **else**
22:             **if** $Nsol.cost = 0$ **then**
23:                 Add $Nsol$ to the $BestL$.
24:             **else**
25:                 let $C \leftarrow Nsol$, $count \leftarrow count + 1$ and go to the line 8.
26:             **end if**
27:         **end if**
28:     **end while**
29: **end if**
30: Return the best solution in $BestL$. //Return element in $BestL$ with the fewest conflicts.

5. Experimental Results

A fullerene graph is a planar 3-connected cubic graph whose faces are pentagons and hexagons. In this section, our aim is to determine the recognition number

Algorithm 2 Calculating the number of conflicts in a coloring (CA)

Require: An arbitrary coloring, $C.value$, the number of graph's vertices (n), the number of colors (k).

Ensure: The number of conflicts for $C.value$, ($conf$).

1: Generate the coding matrix CM, which is a $n \times (k+1)$ matrix that each of its rows denotes the code of a vertex.

2: Let $conf \leftarrow 0$.

3: Compare the rows of matrix CM pairwise, and if they are equal let $conf \leftarrow conf + 1$.

4: Return $conf$.

of some fullerene graphs and explain the application of the algorithm presented in the previous section. For this purpose, we have implemented Algorithm 1, which is coded in Python 3.11.5 64-bit (CPU Intel(R) Core(TM) i5-4300M CPU @ 2.60GHz 2.60 GHz). Descriptions of the sample graphs are provided in Sections 5.1 to 5.3, and Table 1 presents the recognition number obtained by the algorithm ($LSArn$), the bound obtained from Theorem 5.1 (rnb), the number of iterations required for the algorithm to reach the result ($NIter$), and the CPU time. For all graphs, the maximum number of iterations was set to 500 ($Maxit = 500$), and the iterations reported in the table indicate the iteration at which the number of conflicts reached zero ($conf = 0$) and the algorithm terminated. It is clear that the number of vertices in the graph, and consequently the number of neighborhoods that need to be examined, as well as the number of colors, are factors that affect the number of algorithm iterations and CPU time.

Before reporting the obtained results, we recall the following theorem.

Theorem 5.1 [1]**.** *If c is a k-coloring of the vertices of a graph G, then the number of different possible color codes of the vertices of degree r in G is $k\binom{r+k-1}{r}$.*

By replacing $r = 3$ in Theorem 5.1, we have the below relation for a fullerene graph G

$$n(G) \leqslant \frac{k^4 + 3k^3 + 2k^2}{6}. \tag{5.1}$$

Therefore, to find the recognizable coloring of the desired fullerene graph using Algorithm 1, it suffices to find the minimum value of k that satisfies Theorem 5.1, as input to the algorithm.

5.1. Recognizable coloring for dodecahedral graph

Given Theorem 5.1, and considering the degree of vertices in the dodecahedral graph Γ, which are all of degree 3, the number of distinct codes with 2 colors

$(k = 2)$ is not more than 8. In other words, if the number of vertices in the graph exceeds 8, it is not possible to color it with only 2 colors. Therefore, $rn(\Gamma) \geqslant 3$.

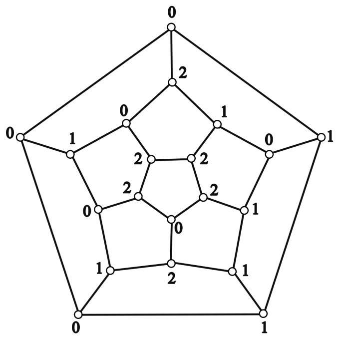

Figure 3. Dodecahedral graph.

A recognizable coloring with 3 colors for this graph is obtained by Algorithm 1 and can be observed in Figure 3.

5.2. Recognizable coloring for $(BN)_{16}$

According to Theorem 5.1, for a graph with vertices of degree 3 and $k = 3$ colors, it is possible to construct 30 distinct codes. This implies that if the number of vertices exceeds 30, it is not possible to have a recognizable coloring for the graph with 3 colors. Since the number of vertices in $(BN)_{16}$ is 32, then $rn((BN)_{16}) \geqslant 4$. A recognizable coloring with 4 colors for this graph can be observed in Figure 4.

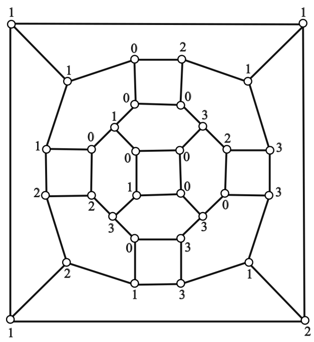

Figure 4. $(BN)_{16}$ graph.

5.3. More examples of fullerene graphs

A fullerene graph is a 3-regular graph, for which we can find the smallest value of k satisfying Theorem 5.1 and use it as the input for Algorithm 1. Figure 5 represents the results obtained from Algorithm 1 for some fullerene graphs.

Graph name	$LSArn$	rnb	$NIter$	$CPUtime(sec)$
$dodecahedral$	3	3	255	2.40
$(BN)_{16}$	4	4	73	1.80
$(12,2)-Generalizedpeterson$	4	3	15	0.24
$26-fullerene$	4	3	49	0.90
$(3,6)-fullerene$	4	3	41	0.96
$Truncatedicosahedral$	6	4	22	2.08
$70-fullerene$	6	4	185	22.45

Table 1. The performance of LSA for sample graphs.

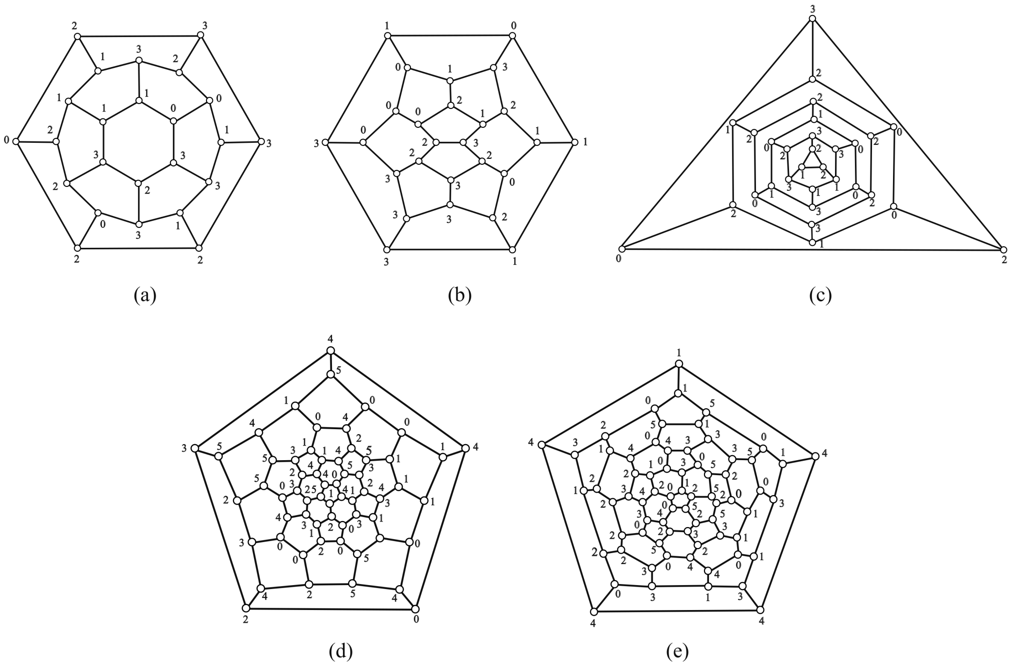

Figure 5. (a) (12,2)-Generalized Peterson graph (b) 26-fullerene graph (c) (3,6)-fullerene $(F)_{6k}$ (d) Truncated icosahedral graph (e) 70-fullerene graph 4085.

In general, for determining the parameter k as an input to Algorithm 1, it is clear that for each graph G, $2 \leqslant k \leqslant n_{\max}$, where if n_i denotes the number of vertices with degree i then $n_{\max} = \max\{n_i : 1 \leqslant i \leqslant \triangle(G)\}$. In the worst-case, we can try the algorithm for various values of $2 \leqslant k \leqslant n_{\max}$.

6. CONCLUSION

In this paper, following the definition of recognizable coloring, we attempt to specifically articulate this type of coloring for corona product graphs. Additionally, a local search algorithm is presented for finding recognizable coloring of any graph G. As an application of the proposed algorithm, recognizable coloring of some fullerene graphs is achieved using this algorithm.

Acknowledgements

The authors would like to thank all the reviewers for their insightful and valuable comments, which led to significant improvements in the manuscript. Research of the second author, Michael A. Henning, supported in part by the South African National Research Foundation (grants 132588, 129265) and the University of Johannesburg. Research of the third author, Mostafa Tavakoli, supported in part by the Ferdowsi University of Mashhad.

References

[1] G. Chartrand, L. Lesniak, D.W. VanderJagt and P. Zhang, *Recognizable colorings of graphs*, Discuss. Math. Graph Theory **28** (2008) 35–57.
https://doi.org/10.7151/dmgt.1390

[2] M.J. Dorfling and S. Dorfling, *Recognizable colorings of cycles and trees*, Discuss. Math. Graph Theory **32** (2012) 81–90.
https://doi.org/10.7151/dmgt.1587

[3] F. Harary and M. Plantholt, *The point-distinguishing chromatic index*, in: Graphs and Applications (Wiley, New York, 1985) 147–162.

[4] T.W. Haynes, S.T. Hedetniemi and M.A. Henning, Domination in Graphs: Core Concepts, Springer Monogr. Math. (Springer, Cham, 2023).
https://doi.org/10.1007/978-3-031-09496-5

[5] Y. Hou and W.-C. Shiu, *The spectrum of the edge corona of two graphs*, Electron. J. Linear Alg. **20** (2010) 586–594.
https://doi.org/10.13001/1081-3810.1395

[6] S. Klavžar and M. Tavakoli, *Dominated and dominator colorings over* (*edge*) *corona and hierarchical products*, Appl. Math. Comput. **390** (2021) 125647.
https://doi.org/10.1016/j.amc.2020.125647

[7] M. Radcliffe and P. Zhang, *Irregular colorings of graphs*, Bull. Inst. Combin. Appl. **49** (2007) 41–59.

Received 5 April 2024
Revised 10 December 2024
Accepted 10 December 2024
Available online 15 January 2025