Packing chromatic number of unitary Cayley graphs of $\mathbb{Z}_n$ and algorithmic approaches to it

Zahra Hamed-Labbafian [a,*] Mostafa Tavakoli [a,†]
Mojgan Afkhami [b,‡] Sandi Klavžar [c,d,e,§]

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

[b] Department of Mathematics, University of Neyshabur, P.O.Box 91136-899, Neyshabur, Iran

[c] Faculty of Mathematics and Physics, University of Ljubljana, Slovenia

[d] Institute of Mathematics, Physics and Mechanics, Ljubljana, Slovenia

[e] Faculty of Natural Sciences and Mathematics, University of Maribor, Slovenia

Abstract

A packing k-coloring of a graph G is a partition of $V(G)$ into k disjoint non-empty classes $V_1, \ldots, V_k$, such that if $u, v \in V_i$, $i \in [k]$, $u \neq v$, then the distance between u and v is greater than i. The packing chromatic number of G is the smallest integer k which admits a packing k-coloring of G. In this paper, the packing chromatic number of the unitary Cayley graph of $\mathbb{Z}_n$ is computed. Two metaheuristic algorithms for calculating the packing chromatic number are also proposed.

Keywords: packing chromatic number; unitary Cayley graph; genetic algorithm; local search algorithm

AMS Subj. Class. (2020): 05C15, 05C85

[*]Email: `hamedlabbafianzahra@gmail.com`
[†]Corresponding author, Email: `m_tavakoli@um.ac.ir`
[‡]Email: `Afkhami@neyshabur.ac.ir`
[§]Email: `sandi.klavzar@fmf.uni-lj.si`

1 Introduction

Packing coloring is a variant of coloring with applications in frequency assignment, resource allocation, and wireless network design. In this variant, vertices of a graph are assigned colors such that vertices sharing the same color are separated by a distance determined by the color itself. More precisely, a *packing k-coloring* of a graph $G = (V(G), E(G))$ is a partition of $V(G)$ into k disjoint non-empty color classes $V_1, \ldots, V_k$, such that V_i, $i \in [k]$, is an *i-packing*, that is, for each two distinct vertices $u, v \in V_i$ we have $\mathrm{d}_G(u, v) \geq i + 1$. The smallest integer k which admits a packing k-coloring of G is the *packing chromatic number* $\chi_\rho(G)$ of G.

The packing chromatic number was initially explored under the name broadcast chromatic number by Goddard et al. in [12]. The terminology and notation used today was proposed in [5]. This coloring concept has already been extensively and deeply researched, and the review article [4] published in 2020 contains 68 references. Research continued with recent studies of the packing chromatic number of iterated Mycielskians [2] and hypercubes [14], and with investigations of variants such as distance dominator packing coloring [9], partial and quasi-packing packing coloring [15], and Grundy packing coloring [13]. Several recent papers deal also with criticality concepts, cf. [10, 18]. The greatest emphasis in recent times, however, has been on S-packing colorings, especially on subcubic graphs, see [3, 20–23, 26].

It should be stressed that it is intrinsically difficult to determine the packing chromatic number. It was proved already in the seminal paper [12] that the decision problem whether an input graph admits a packing k-coloring is NP-complete for $k = 4$, even when restricted to planar graphs. This finding was followed up by Fiala and Golovach [11] with the breakthrough result asserting that this decision problem is is NP-complete for trees. Furthermore, it was later proven that the decision problem remains NP-complete when restricted to chordal graphs with diameter at least 3 [17].

For these reasons, it is desirable to find different heuristic and/or approximation algorithmic approaches to the packing chromatic number. We do this in Section 3 by considering the local search and the genetic algorithm for computing the packing chromatic number. In Section 4 we then report experimental results on these two approaches and make their comparison with the recently proposed greedy approach [13]. Before turning our attention to algorithms, we determine in Section 2 the packing chromatic number of unitary Cayley graphs of $\mathbb{Z}_n$.

In the reminder of this section we recall some definitions and notations, for undefined terms we refer to [25]. Let $G = (V(G), E(G))$ be a graph. We use the notation $x \sim_G y$ to denote that $xy \in E(G)$. A subset S of $V(G)$ is *independent* if no two vertices of S are adjacent. The cardinality of a largest independent set is the *independence number* $\alpha(G)$ of G. The *distance* $\mathrm{d}_G(a, b)$ between vertices a and b of a connected graph G is the length of a shortest a, b-path. The diameter $\mathrm{diam}(G)$ of

G is the length of a longest shortest path in G. By $K_{n_1,\ldots,n_m}$ we denote the complete m-partite graphs with parts of size n_i, $i \in [m]$. The *direct product* $G_1 \times G_2$ of graphs G_1 and G_2 is the graph with vertex set $V(G_1) \times V(G_2)$ and $(u_1, v_1) \sim_{G_1 \times G_2} (u_2, v_2)$ if $u_1 \sim_{G_1} u_2$ and $v_1 \sim_{G_1} v_2$.

2 Packing chromatic number of unitary Cayley graphs of $\mathbb{Z}_n$

In this section we determine the packing chromatic number of the unitary Cayley graph of $\mathbb{Z}_n$, the ring of integers modulo n.

Let R be a finite commutative ring with nonzero identity, and let $R^\times$ denote the set of all unit elements of R. The *unitary Cayley graph* $G_R = \mathrm{Cay}(R, R^\times)$ of R is a graph with the vertex set R and two vertices x and y are adjacent if $x - y \in R^\times$. We refer to [6, 7, 24] for some recent investigations of unitary Cayley graphs, see also [1, 16, 19].

If R is a finite commutative ring, then by [8, p. 752] we can write $R \cong R_1 \times \cdots \times R_t$, where R_i, $i \in [t]$, is a finite local ring with maximal ideal $\mathfrak{m}_i$. This decomposition is unique up to permutation of factors. We denote the (finite) residue field $\frac{R_i}{\mathfrak{m}_i}$ by K_i and $f_i = |K_i| = \frac{|R_i|}{|\mathfrak{m}_i|}$. We also assume (after appropriate permutation of factors) that $f_1 \leqslant \cdots \leqslant f_t$.

The following proposition is a basic consequence of the definition of the unitary Cayley graphs, cf. [1, Proposition 2.2].

Proposition 2.1. *If R is a finite commutative ring, then the following statements hold.*

(a) The graph G_R is a $|R^\times|$-regular graph.

(b) If R is a local ring with maximal ideal $\mathfrak{m}$, then G_R is a complete multipartite graph whose partite sets are the cosets of $\mathfrak{m}$ in R. In particular, G_R is a complete graph if and only if R is a field.

(c) If $R \cong R_1 \times \cdots \times R_t$ is a product of local rings, then $G_R \cong \times_{i=1}^{t} G_{R_i}$. Hence, G_R is the direct product of complete multipartite graphs.

Proposition 2.2. [1, Theorem 3.1] *If $R \cong R_1 \times \cdots \times R_t$ is a finite, commutative ring, then*

$$\mathrm{diam}(G_R) = \begin{cases} 1; & t = 1 \text{ and } R \text{ is a field,} \\ 2; & t = 1 \text{ and } R \text{ is not a field,} \\ 2; & t \geqslant 2, f_1 \geqslant 3, \\ 3; & t \geqslant 2, f_1 = 2, f_2 \geqslant 3, \\ \infty; & t \geqslant 2, f_1 = f_2 = 2. \end{cases}$$

For the rest of this section, some preparation is needed. Let $n = p_1^{r_1} \dots p_t^{r_t}$ be the prime factorization of n, where p_is are primes with $p_1 < \cdots < p_t$. Then $\mathbb{Z}_n \cong \mathbb{Z}_{p_1^{r_1}} \times \cdots \times \mathbb{Z}_{p_t^{r_t}}$. Also $\mathbb{Z}_{p_i^{r_i}}$ is a local ring with the maximal ideal $\mathfrak{m}_i = \{rp_i \mid r \in \mathbb{Z}_{p_i^{r_i}}\}$ with $|\mathfrak{m}_i| = p_i^{r_i - 1}$ and the number of cosets of $\mathfrak{m}_i$ in $\mathbb{Z}_{p_i^{r_i}}$ is equal to p_i, for each $i \in [t]$.

Remark 2.3. *By Proposition 2.1 and the above preparation,* $G_{\mathbb{Z}_{p_i^{r_i}}}$, $i \in [t]$, *is isomorphic to the complete* p_i*-partite graph* $K_{p_i^{r_i-1}, \dots, p_i^{r_i-1}}$. *Also, since* $\mathbb{Z}_n \cong \mathbb{Z}_{p_1^{r_1}} \times \cdots \times \mathbb{Z}_{p_t^{r_t}}$, *the third part of Proposition 2.1, yields*

$$G_{\mathbb{Z}_n} \cong \times_{i=1}^{t} K_{\underbrace{p_i^{r_i-1}, \dots, p_i^{r_i-1}}_{p_i}}.$$

Using Proposition 2.2, the following result can be deduced.

Corollary 2.4. *If* $p_1^{r_1} \dots p_t^{r_t}$ *is the prime factorization of* n, *where* $p_1 < \cdots < p_t$, *then* $G_{\mathbb{Z}_n}$ *is connected and we have*

$$\operatorname{diam}(G_{\mathbb{Z}_n}) = \begin{cases} 1; & t = 1, r_1 = 1, \\ 2; & t = 1, r_1 > 1, \\ 2; & t \geqslant 2, p_1 \geqslant 3, \\ 3; & t \geqslant 2, p_1 = 2. \end{cases}$$

Now all is ready to determine the packing chromatic number of $G_{\mathbb{Z}_n}$. The case when n is a prime is trivial since in that case $G_{\mathbb{Z}_n}$ is a complete graph. By Corollary 2.4 the remaining cases are of diameter two or three. For the first case we recall the following result from the seminal paper [12], see also the survey [4, Proposition 2.5].

Proposition 2.5. *If* G *is a graph, then*

$$\chi_\rho(G) \le n(G) - \alpha(G) + 1,$$

with equality if $\operatorname{diam}(G) = 2$.

Assume that $n = p^r$, where p is a prime number and $r > 1$. In this situation, by Corollary 2.4, we have $\operatorname{diam}(G_{\mathbb{Z}_{p^r}}) = 2$, and so by Proposition 2.5, we have $\chi_\rho(G_{\mathbb{Z}_{p^r}}) = p^r - \alpha(G_{\mathbb{Z}_{p^r}}) + 1$. By Remark 2.3, $G_{\mathbb{Z}_{p^r}}$ is isomorphic to the p-partite $K_{p^{r-1}, \dots, p^{r-1}}$, for which we have $\alpha(G_{\mathbb{Z}_{p^r}}) = p^{r-1}$. Therefore, if $n = p^r$, $r > 1$, then

$$\chi_\rho(G_{\mathbb{Z}_n}) = p^r - p^{r-1} + 1.$$

The second subcase when the diameter of the unitary Cayley graph is two in covered with the next result.

Theorem 2.6. *If $n = p_1^{r_1} \dots p_t^{r_t}$, where $t > 1$ and $3 \leqslant p_1 < \dots < p_t$, then*

$$\chi_\rho(G_{\mathbb{Z}_n}) = p_1^{r_1-1} p_2^{r_2} \dots p_t^{r_t}(p_1 - 1) + 1.$$

Proof. By Corollary 2.4, $\operatorname{diam}(G_{\mathbb{Z}_n}) = 2$. Hence by Proposition 2.5 it suffices to determine $\alpha(G_{\mathbb{Z}_n})$. By Remark 2.3, $G_{\mathbb{Z}_n}$ is isomorphic to the direct product of t p_i-partite graphs $K_{p_i^{r_i-1},\dots,p_i^{r_i-1}}$, $i \in [t]$. Also, $G_{\mathbb{Z}_{p_i}}$, $i \in [t]$, is isomorphic to the p_i-partite graph $K_{p_i^{r_i-1},\dots,p_i^{r_i-1}}$. Let $V_1^{p_i}, \dots, V_{p_i}^{p_i}$ be the p_i parts of $G_{\mathbb{Z}_{p_i}}$, and $V_j^{p_i} = \{a_{j,1}^{p_i}, \dots, a_{j,p_i^{r_i-1}}^{p_i}\}$, $j \in [p_i]$. Clearly $|V_j^{p_i}| = |V_{j'}^{p_i}| = p_i^{r_i-1}$, for each $j, j' \in [p_i]$. For $j \in [p_i]$ set

$$A_j^{p_i} = V(G_{\mathbb{Z}_{p_1}}) \times \dots \times V(G_{\mathbb{Z}_{p_{i-1}}}) \times V_j^{p_i} \times V(G_{\mathbb{Z}_{p_{i+1}}}) \times \dots \times V(G_{\mathbb{Z}_{p_t}}).$$

The set $A_j^{p_i}$, $j \in [p_i]$, is an independent set of order $p_1^{r_1} \dots p_{i-1}^{r_{i-1}} p_i^{r_i-1} p_{i+1}^{r_{i+1}} \dots p_t^{r_t}$, and

$$V(G_{\mathbb{Z}_n}) = \bigcup_{i=1}^{t} \bigcup_{j=1}^{p_i} A_j^{p_i}.$$

Since $p_1 < \dots < p_t$ and $|\bigcup_{j=1}^{p_1} A_j^{p_1}| = \dots = |\bigcup_{j=1}^{p_t} A_j^{p_t}|$, we have $|A_1^{p_1}| > \dots > |A_1^{p_t}|$.

Let S be an arbitrary independent set of $G_{\mathbb{Z}_n}$. Since there are at most $p_1^{r_1-1}$ possibilities for the first component of the elements of S, we must have $|S| \leqslant p_1^{r_1-1} p_2^{r_2} \dots p_t^{r_t}$, which means that $|S| \leqslant |A_1^{p_1}|$. Therefore, $A_1^{p_1}$ is an independent set of greatest order, which means that $\alpha(G_{\mathbb{Z}_n}) = |A_1^{p_1}| = p_1^{r_1-1} p_2^{r_2} \dots p_t^{r_t}$. Proposition 2.5 completes the argument. □

In view of Corollary 2.4, the only remaining case to be considered is the following.

Theorem 2.7. *If $n = p_1^{r_1} \dots p_t^{r_t}$, where $t > 1$, $p_1 = 2$, and $p_1 < \dots < p_t$, then*

$$\chi_\rho(G_{\mathbb{Z}_n}) = p_1^{r_1-1} p_2^{r_2} \dots p_t^{r_t}(p_1 - 1).$$

Proof. By the proof of Theorem 2.6, the greatest size of a 1-packing of $G_{\mathbb{Z}_n}$ is $p_1^{r_1-1} p_2^{r_2} \dots p_t^{r_t}$. By Corollary 2.4, each two distinct vertices of $G_{\mathbb{Z}_n}$ are at distance at most three. Also, by Remark 2.3, we have $G_{\mathbb{Z}_{2^{r_1}}} \cong K_{2^{r_1-1},2^{r_1-1}}$. Let V_1 and V_2 be the two parts of $G_{\mathbb{Z}_{2^{r_1}}}$. If $X = (x_1, \dots, x_t)$ and $Y = (y_1, \dots, y_t)$ are nonadjacent vertices of $G_{\mathbb{Z}_n}$, then $\mathrm{d}_{G_{\mathbb{Z}_n}}(X, Y) = 3$ if and only if $x_1 \in V_1$ and $y_1 \in V_2$, or vice versa. Now let A be a 2-packing in $G_{\mathbb{Z}_n}$. If $|A| > 2$, then there are three distinct vertices $U_1 = (x_1, \dots, x_t)$, $U_2 = (y_1, \dots, y_t)$, and $U_3 = (z_1, \dots, z_t)$ in A such that $\mathrm{d}_{G_{\mathbb{Z}_n}}(U_i, U_j) = 3$, for each $1 \leqslant i \neq j \leqslant 3$. Without loss of generality, assume that $x_1 \in V_1$ and $y_1 \in V_2$. If $z_1 \in V_1$, then we have $\mathrm{d}_{G_{\mathbb{Z}_n}}(U_1, U_3) = 2$, which is impossible. If $z_1 \in V_2$, then we have $\mathrm{d}_{G_{\mathbb{Z}_n}}(U_2, U_3) = 2$, which is again impossible. So any 2-packing in $G_{\mathbb{Z}_n}$ has at most two vertices and for $i > 2$, each i-packing of $G_{\mathbb{Z}_n}$ has at most one vertex. We can conclude that $\chi_\rho(G_{\mathbb{Z}_n}) = p_1^{r_1-1} p_2^{r_2} \dots p_t^{r_t}(p_1 - 1)$. □

3 Two metaheuristic algorithms

Recall from the introduction that there is a strong case for utilizing meta-heuristic algorithms for the approximation of the packing chromatic number. In this section we employ the local search and the genetic algorithm for solving this problem. We recall that the genetic algorithm is designed as a population-centric technique, whereas the local search algorithm focuses on point-based optimization.

3.1 Local search algorithm

We now present a local search (LS) algorithm for finding a packing k-coloring for a given graph G.

The main algorithm's objective is to achieve a packing k-coloring by iteratively improving an initial solution. $Maxit$ in Algorithm 1 denotes the maximum number of iterations chosen as a stopping criterion. In addition, for given k the notation c_{fitness} is an evaluation function for a coloring c which is defined as $c_{\text{fitness}} = \frac{1}{1+I_c}$ where $I_c = \sum_{i=1}^{k} |\{\{u, v\} \subset V(G) : \; u \neq v, c(u) = c(v) = i, \mathrm{d}_G(u, v) \leq i\}|$. This process is performed by the sub-algorithm named the fitness calculation (FC) algorithm (FC for short) for each coloring. Algorithm 1 continues by generating neighboring solutions. If for a specific k, a neighbor solution c^i is found such that $c^i_{\text{fitness}} = 1$, we have a packing k-coloring. Otherwise, another value of k can be checked. Moreover, N_c denotes a set of neighboring configurations, each of then differ from the current coloring c in only one vertex's color.

3.2 Detailed explanation of the LS algorithm

Initialization

We start by generating a random coloring c of $V(G)$ with k colors. Then we apply the FC sub-algorithm to compute the fitness number c_{fitness}. If $c_{\text{fitness}} = 1$, it's immediately returned as the best solution.

Local search loop

If $c_{\text{fitness}} < 1$, then the algorithm enters a local search loop. In each iteration, a set of neighboring configurations N_c is generated. For each neighbor configuration of N_c, the algorithm computes the fitness function and updates the current solution if a neighboring solution with grater fitness value is found.

Updating best solution

If the fitness value of the best neighboring solution is greater than c_{fitness}, then we update S_{best}; In other word, If $f_{\text{N}} > f_{\text{best}}$, the algorithm sets $S_{\text{best}} = N_{\text{best}}$ and $f_{\text{best}} = f_{\text{N}}$. We recall that if $f_{\text{best}} = 1$, then the algorithm returns S_{best} as the best solution. Otherwise, the algorithm increments the iteration count.

Termination

The process continues until either a coloring c with $c_{\text{fitness}} = 1$ is found, the maximum number of iterations (Maxit) is reached, or no further improvement can be made. Finally, the algorithm returns S_{best} as the best solution.

Algorithm 1 LS algorithm to find a packing k-coloring

Require: Graph, G, the number of colors k, the number of iterations $Maxit$.
Ensure: A packing k-coloring of G.

Generate an arbitrary vertex coloring, c, for G.
Call FC algorithm for c ($c_{\text{fitness}} \leftarrow FC(G, k, c)$).
Set $S_{\text{best}} = c$ and $f_{\text{best}} = c_{\text{fitness}}$.
if $f_{\text{best}} = 1$ **then**
    Return S_{best}.
else
    Let $count \leftarrow 1$. //count is the iteration counter.
    while $count <= Maxit$ **do**
        Generate neighborhood set N_c of c.
        Set $N_{\text{best}} = \emptyset$ and $f_{\text{N}} = 0$.
        while $N_c \neq \varnothing$ **do**
            Select an element, c^i of N_c and call FC for it ($c^i_{\text{fitness}} \leftarrow FC(G, k, c^i)$).
            if $c^i_{\text{fitness}} > f_{\text{N}}$ **then**
                $N_{\text{best}} \leftarrow c^i$ and $f_{\text{N}} \leftarrow c^i_{\text{fitness}}$.
            end if
            $N_c \leftarrow N_c - \{c^i\}$.
        end while
        if $f_{\text{N}} > f_{\text{best}}$ **then**
            Set $S_{\text{best}} = N_{\text{best}}$ and $f_{\text{best}} = f_{\text{N}}$.
            if $f_{\text{best}} = 1$ **then**
                Return S_{best} and break.
            end if
        end if
        Set $c \leftarrow N_{\text{best}}$, $count \leftarrow count + 1$.
    end while
end if
Return S_{best}.

Algorithm 2 FC algorithm

Require: graph G, number of colors k, coloring c.

Ensure: The fitness value c_{fitness}.

```
Set I_c = 0.
for i = 1 to k do
    for each u, v ∈ V(G) such that c(u) = c(v) = i do
        if d_G(u, v) ≤ i then
            Set I_c ← I_c + 1.
        end if
    end for
end for
Set c_fitness = 1/(1+I_c).
Return c_fitness.
```

3.3 Genetic algorithm

Here we apply the genetic algorithm for solving the packing coloring problem. This algorithm begins by generating an initial population of size n_p, and calculating its fitness values via algorithm FC. If a solution achieves the fitness value 1, then it is immediately returned as the optimal solution. Otherwise, the algorithm proceeds to the main iterative phase.

At the beginning of each iteration, offspring and mutated populations are generated through crossover and mutation. These populations are merged with the original, and less optimal solutions are removed to create a refined population of size n_p. This process is repeated until a maximum number of iterations is reached. Finally, the algorithm selects the solution with the highest fitness value. If the fitness equals 1, it outputs a graph coloring. Otherwise, the algorithm reports that it was unable to color the graph using k colors. Below, we describe the combination operations employed by our algorithm to generate a new population.

Crossover 1

In this process of generating a new offspring, two parents are randomly selected from the existing population. The resulting offspring is initially created as an exact copy of the second parent's genome. Subsequently, a specific modification mechanism is applied: any colors present in the first parent's genetic structure but absent in the second parent are randomly substituted into selected vertices of the offspring's structure.

Crossover 2

The process of generating new offspring begins with the random selection of two parents from the population. The initial offspring structure is formed through complete replication of the first parent's genome. Subsequently, in a targeted modification process, two vertices are randomly selected from the offspring's structure and their corresponding colors are transferred from the second parent. This genetic transfer mechanism ensures a balanced combination of characteristics from both parents.

Mutation
The process begins with random selection of a parent from the existing population. The mutated's initial structure is established through complete replication of the parent's genome. Subsequently, in a targeted modification process, two vertices are randomly selected from the mutated's structure, and their colors are interchanged. This mutation mechanism enhances genetic diversity within the population and facilitates the exploration of new regions in the search space.

GA is presented as Algorithm 3.

Algorithm 3 GA for the packing coloring problem

Require: G, k, maximum number of iterations $Maxit$, initial population's size n_p.
Ensure: a k-packing coloring for G.

Create an initial populations with size n_p, and evaluate them using FC algorithm.
$count = 1$.
while $count \leq Maxit$ **do**
    Select parents randomly.
    Generate and evaluate offspring using the crossover operators.
    Generate and evaluate mutated populations using the mutation operator.
    Merge the initial population, offspring, and mutated populations, and sort them based on the fitness function (new population).
    Truncate new population and generate a new population with size n_p.
    $count = count + 1$.
end while
Return the individual with the best fitness value as the best solution.

4 Experimental results

The section includes two parts: the initial part reviews the efficiency of GA, while the latter part delivers a comparative analysis of the three algorithms: LS, GA, and Greedy. In what follows, pc_{GA} and pc_{Greedy} refer to the approximations of the packing chromatic number computed by the genetic algorithm and the greedy algorithm, respectively.

4.1 GA efficiency

Here we focus on assessing the effectiveness of the earlier described GA. We first present in Table 1 a report on the algorithm's output for graphs with known exact chromatic numbers as provided in [4].

Table 1: χ_ρ of some graphs computed by GA

Graph name	χ_ρ	pc_{GA}	GA CPU time (s)
C_{15}	4	4	0.458
C_{20}	3	3	3.442
P_{20}	3	3	5.721
S_{10}	2	2	0.120
$K_{3,5,7}$	9	9	0.372

GA has been next tested on some examples from Section 2, see Table 2.

Table 2: χ_ρ of some unitary Cayley graphs computed by GA

Graph name	χ_ρ	pc_{GA}	GA CPU time (s)
$G_{\mathbb{Z}_5}$	5	5	0.002
$G_{\mathbb{Z}_{16}}$	9	9	0.299
$G_{\mathbb{Z}_{21}}$	15	15	2.103
$G_{\mathbb{Z}_{27}}$	19	19	1.893
$G_{\mathbb{Z}_{45}}$	31	31	22.196

4.2 Comparison of the three algorithms

Article [13] introduces a greedy algorithm designed to assist in determining the chromatic number of graphs. We compared the results obtained from both the greedy and genetic algorithms. Through an analysis of various graph samples, particularly a subset of Cayley graphs, we found that for some graphs, all three methods found packing colorings of the same cardinality. Some of these examples are grouped together in Table 3.

Although LS is faster than GA, it might get stuck in a local maximum and miss a better solution. Hence we moved on to compare the genetic and the greedy algorithm on some larger graphs. In Table 4 computational results are collected for

Table 3: Graphs for which all three algorithms return the same value

Graph	value	Greedy CPU time (s)	GA CPU time (s)	LS CPU time (s)
$\mathrm{Cay}(\mathbb{Z}_8, \{1,3,5,7\})$	5	0.001	0.039	0.004
$\mathrm{Cay}(\mathbb{Z}_9, \{1,3,5,7\})$	7	0.002	0.029	0.007
$\mathrm{Cay}(\mathbb{Z}_{12}, \{1,3,9,11\})$	7	0.002	0.769	0.023
$\mathrm{Cay}(\mathbb{Z}_8, \{1,2,3,5,6,7\})$	7	0.001	0.022	0.002
$\mathrm{Cay}(\mathbb{Z}_9, \{1,2,3,6,7,8\})$	8	0.002	0.020	0.006
$\mathrm{Cay}(\mathbb{Z}_{12}, \{1,2,3,9,10,11\})$	10	0.002	0.054	0.017

the generalized Petersen graph $G(12, 2)$, the graph BN16 (shown in 1), the truncated icosahedral graph TI and the fullerene graphs C_{48} and C_{70}.

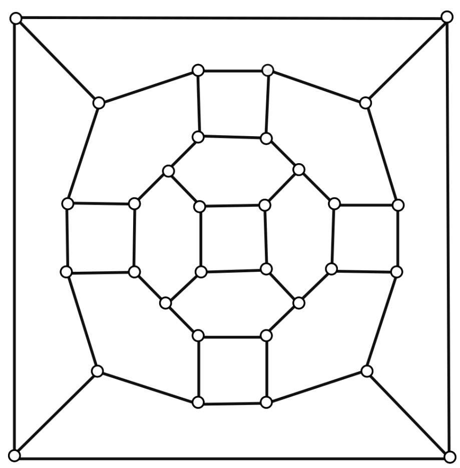

Figure 1: Graph BN16

A comparative analysis of the greedy and genetic algorithm reveals that as graph complexity and size increase, the performance gap between these methods becomes more pronounced. For instance, for the fullerene C_{48}, the greedy algorithm yielded inconsistent results across multiple iterations (e.g., 16, 17, 16, 17, 19, 18), highlighting its instability and inefficiency for complex graphs. On the other hand, GA found a packing coloring with 13 colors, but using for it 443.667 seconds. Nevertheless, GA demonstrated its ability to explore a broader solution space and identify a better, most likely optimal result.

These insights offer guidance for selecting graph coloring strategies in future research, especially for large and complex graphs. If an upper bound for the packing chromatic number can be established using some of the results from [4] or elsewhere, it can serve as input for the genetic algorithm; otherwise, the greedy algorithm can

Table 4: Computational results for some additional graphs

Graph G	$n(G)$	pc_{Greedy}	Greedy CPU time (s)	pc_{GA}	GA CPU time (s)
$G(12,2)$	24	13,11,13, 12,13,12	0.007	10	60.343
BN16	32	14,14,13, 13,11,12	0.011	9	89.511
C_{48}	48	17,16,17, 16,19,18	0.028	13	443.667
TI	60	21,21,21, 17,19,23	0.055	16	549.806
C_{70}	70	22,22,26, 21,24,25	0.081	19	1237.402

provide an initial estimate, which can then be refined by the genetic algorithm.

Acknowledgments

Sandi Klavžar were supported by the Slovenian Research and Innovation Agency (ARIS) under the grants P1-0297, N1-0355, and N1-0285. The research of the second author, Mostafa Tavakoli, was supported in part by the Ferdowsi University of Mashhad.

Declaration of interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Our manuscript has no associated data.

References

[1] R. Akhtar, M. Boggess, T. Jackson-Henderson, I. Jiménez, R. Karpman, A. Kinzel, D. Pritikin, On the unitary Cayley graph of a finite ring, Electron. J. Combin. 16 (2009) R117.

[2] E. Bidine, T. Gadi, M. Kchikech, The exponential growth of the packing chromatic number of iterated Mycielskians, Discrete Appl. Math. 341 (2023) 232–241.

[3] B. Brešar, J. Ferme, P. Holub, M. Jakovac, P. Melicharová, S-packing colorings of distance graphs with distance sets of cardinality 2, Appl. Math. Comput. 490 (2025) 129200.

[4] B. Brešar, J. Ferme, S. Klavžar, D.F. Rall, A survey on packing colorings, Discuss. Math. Graph Theory 40 (2020) 923–970.

[5] B. Brešar, S. Klavžar D.F. Rall, On the packing chromatic number of Cartesian products, hexagonal lattice, and trees, Discrete Appl. Math. 155 (2007) 2303–2311.

[6] A. Burcroff, Domination parameters of the unitary Cayley graph of $\mathbb{Z}/n\mathbb{Z}$, Discuss. Math. Graph Theory 43 (2023) 95–114.

[7] D. Dolžan, The unitary Cayley graph of a semiring, Discrete Math. 347 (2024) 113873.

[8] D.S. Dummit, R.M. Foote, Abstract Algebra, Third Edition, John Wiley & Sons, Hoboken, NJ, 2004.

[9] J. Ferme, D. Štesl, On distance dominator packing coloring in graphs, Filomat 35 (2021) 4005–4016.

[10] J. Ferme, A characterization of 4-χ_ρ-(vertex-)critical graphs, Filomat 36 (2022) 6481–6501.

[11] J. Fiala, P.A. Golovach, Complexity of the packing coloring problem for trees, Discrete Appl. Math. 158 (2010) 771–778.

[12] W. Goddard, S.M. Hedetniemi, S.T. Hedetniemi, J.M. Harris, D.F. Rall, Broadcast chromatic numbers of graphs, Ars Combin. 86 (2008) 33–49.

[13] D. Gözüpek, I. Peterin, Grundy packing coloring of graphs, Discrete Appl. Math. 371 (2025) 17–30.

[14] P. Gregor, J. Kranjc, B. Lužar, K. Štorgel, Packing coloring of hypercubes with extended Hamming codes, Discrete Appl. Math. 359 (2024) 269–277.

[15] H. Grochowski, K. Junosza-Szaniawski, Partial packing coloring and quasi-packing oloring of the triangular grid, Discrete Math. 348 (2025) 114308.

[16] A. Ilić, The energy of unitary Cayley graphs, Linear Algebra Appl. 431 (2009) 1881–1889.

[17] M. Kim, B. Lidický, T. Masařík, F. Pfender, Notes on complexity of packing coloring, Inf. Process. Lett. 137 (2018) 6–10.

[18] S. Klavžar, H. Lei, X. Lian, Y. Shi, A characterization of 4-χ_S-vertex-critical graphs for packing sequences with $s_1 = 1$ and $s_2 \geq 3$, Discrete Appl. Math. 338 (2023) 46–55.

[19] W. Klotz, T. Sander, Some properties of unitary Cayley graphs, Electron. J. Combin. 14 (2007) #R45.

[20] A. Kostochka, X. Liu, Packing $(1, 1, 2, 4)$-coloring of subcubic outerplanar graphs, Discrete Appl. Math. 302 (2021) 8–15.

[21] R. Liu, X. Liu, M. Rolek, G. Yu, Packing $(1, 1, 2, 2)$-coloring of some subcubic graphs, Discrete Appl. Math. 283 (2020) 626–630.

[22] M. Mortada, About S-packing coloring of 3-irregular subcubic graphs, Discrete Appl. Math. 359 (2024) 16–18.

[23] M. Mortada, O. Togni, Further results and questions on S-packing coloring of subcubic graphs, Discrete Math. 348 (2025) 114376.

[24] J. Rattanakangwanwong, Y. Meemark, Subgraph of unitary Cayley graph of matrix algebras induced by idempotent matrices, Discrete Math. 347 (2024) 113739.

[25] D.B. West, Introduction to Graph Theory, Second Edition, Prentice Hall, Upper Saddle River, NJ, 2001.

[26] W. Yang, B. Wu, On packing S-colorings of subcubic graphs, Discrete Appl. Math. 334 (2023) 1–14.