Mathematics

http://www.aimspress.com/journal/Math

AIMS Mathematics, 9(5): 11644–11659.
DOI:10.3934/math.2024571
Received: 31 January 2024
Revised: 08 March 2024
Accepted: 12 March 2024
Published: 26 March 2024

Research article

On the packing number of 3-token graph of the path graph P_n

Christophe Ndjatchi[1,*], **Joel Alejandro Escareño Fernández**[2], **L. M. Ríos-Castro**[3], **Teodoro Ibarra-Pérez**[4], **Hans Christian Correa-Aguado**[4] **and Hugo Pineda Martínez**[5]

[1] Academia de Físico-Matemáticas, Instituto Politécnico Nacional, UPIIZ, P. C. 098160, Zacatecas, México

[2] Ingeniería Mecatrónica, Becario BEIFI-IPN, Instituto Politécnico Nacional, UPIIZ, P. C. 098160, Zacatecas, México

[3] Academia de Físico-Matemáticas, Instituto Politécnico Nacional, CECYT18, Zacatecas, P. C. 098160, Zacatecas, México

[4] Academia de Ingeniería, Instituto Politécnico Nacional, UPIIZ, P. C. 098160, Zacatecas, México

[5] Unidad Académica de Ingenieria Eléctrica, Universidad Autónoma de Zacatecas, Zacatecas, México

* **Correspondence:** Email: mndjatchi@ipn.mx.

Abstract: In 2018, J. M. Gómez et al. showed that the problem of finding the packing number $\rho(F_2(P_n))$ of the 2-token graph $F_2(P_n)$ of the path P_n of length $n \geq 2$ is equivalent to determining the maximum size of a binary code S' of constant weight $w = 2$ that can correct a single adjacent transposition. By determining the exact value of $\rho(F_2(P_n))$, they proved a conjecture of Rob Pratt. In this paper, we study a related problem, which consists of determining the packing number $\rho(F_3(P_n))$ of the graph $F_3(P_n)$. This problem corresponds to the Sloane's problem of finding the maximum size of S' of constant weight $w = 3$ that can correct a single adjacent transposition. Since the maximum packing set problem is computationally equivalent to the maximum independent set problem, which is an NP-hard problem, then no polynomial time algorithms are expected to be found. Nevertheless, we compute the exact value of $\rho(F_3(P_n))$ for $n \leq 12$, and we also present some algorithms that produce a lower bound for $\rho(F_3(P_n))$ with $13 \leq n \leq 44$. Finally, we establish an upper bound for $\rho(F_3(P_n))$ with $n \geq 13$.

Keywords: packing number; 3-token graphs; error correcting codes; binary codes; algorithms
Mathematics Subject Classification: 05C10, 05C45

1. Introduction

Throughout this paper, $G = (V(G), E(G))$ denotes a finite connected, undirected, and simple (without loops or parallel edges) graph of order $n \geq 3$, where $V(G)$ and $E(G)$ are, respectively, the *vertex set* and *edges set* of G. If $x, y \in V(G)$ and x and y are adjacent, then $\{x, y\} \in E(G)$ and we often write xy instead of $\{x, y\}$. If $k \leq n - 1$ is a positive integer, then the *k-token graph* $F_k(G)$ of G is the graph whose vertices are all the k-subsets of $V(G)$, and two k-subsets A, B are adjacent whenever their symmetric difference $A \triangle B$ defined as $(A \cup B) \setminus (A \cap B)$ is a 2-set $\{a, b\}$ such that $ab \in E(G)$ with $a \in A$ and $b \in B$. As an example of token graphs, see Figure 1 a). The token graphs have been extensively studied, see for instance [3, 4, 6, 7, 12–14]. In those works, problems related to connectivity, diameter, clique number, chromatic number, independence number, Hamiltonian paths, matching number, planarity, regularity, etc. of token graphs have been studied. As the reader can check in [1, 3, 4, 7, 10] and the references therein, the research on token graphs is still of interest.

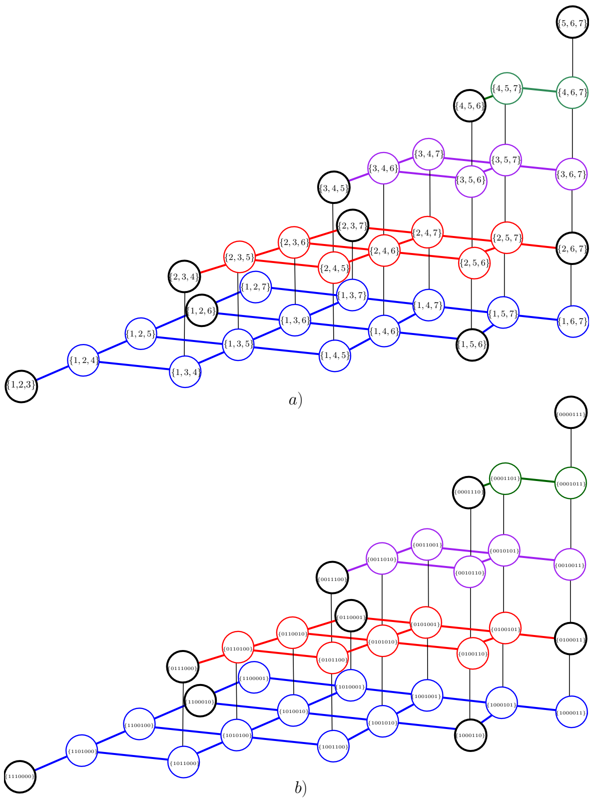

Figure 1. The graph in a) is $F_3(P_7)$, and the graph in b) is the binary code graph Γ_7^3. Clearly, $F_3(P_7)$ and Γ_7^3 are isomorphic. Note that $F_3(P_7)$ and Γ_7^3 can be drawn as a pyramid with 5 floors. The black vertices in $F_3(P_7)$ and Γ_7^3 form a packing set of order 9.

Packing number: The packing number of a graph is a graph invariant which is defined as follows: given a graph G, *the packing number* of G denoted by $\rho(G)$ is the cardinality of a maximum subset S of $V(G)$ such that for each pair of distinct vertices u and v of S, the distance between them is greater than 2. As far as we know, the exact value of the packing number of the k-token graph is known only for $F_2(P_n)$ [19]. In [15], Ríos gave the following lower bound for $\rho(F_3(P_n))$

$$c(n) = \begin{cases} \frac{1}{54}(n^3 + 3n^2) & \text{if } n \equiv 0 \pmod 3, \\ \frac{1}{54}(n^3 + 3n^2 - 4) & \text{if } n \equiv 1 \pmod 3, \\ \frac{1}{54}(n^3 + 3n^2 - 6n - 8) & \text{if } n \equiv 2 \pmod 3. \end{cases} \tag{1.1}$$

In particular, the determination of the exact value of $\rho(F_k(G))$ remains open for $G = P_n$ and $k \geq 3$, and also for $k = 2$ and $G \neq P_n$.

The Neil Sloane's problem [17, 19]. Let n and w be two positive integers such that $0 \leq w \leq n$. We will use $\mathbb{F}_2^n$ to denote the set of all vectors of length n, with entries in $\{0, 1\}$. *A binary code of length n and constant weight w* is a subset S of $\mathbb{F}_2^n$ such that every $u \in S$ has exactly w $1's$ and $n - w$ $0's$. Let $u \in \mathbb{F}_2^n$ and let $N(u)$ be the set of all vectors in $\mathbb{F}_2^n$ which can be obtained from u by transposing a pair of bits [2, 17]. Following the notations in [17, 19], let us define Γ_n^w as the graph whose vertex set is $V(\Gamma_n^w) = S$, so $|V(\Gamma_n^w)| = \binom{n}{w}$, and two vertices $u, v \in V(\Gamma_n^w)$ are adjacent if and only if v can be obtained from u by transposing a pair of adjacent bits, for instance see Figure 1 *b*). Any binary code $S' \subseteq S$ is called a *correcting code* if $N(u) \cap N(v) = \emptyset$ for all $u, v \in S'$ with $u \neq v$. Then S' can correct a single adjacent transposition if and only if S' is a packing set of Γ_n^w. The graph Γ_n^w will be called *a binary code graph of length n and constant weight w*. Neil Sloane's problem consists of determining the maximum cardinality of such a code S', which is equal to $\rho(\Gamma_n^w)$.

In [19], it was shown that the problem of determining $\rho(F_2(P_n))$ is equivalent to finding the maximum code S' of constant weight $w = 2$ which can correct a single adjacent transposition. They computed the exact value of $\rho(F_2(P_n))$ and proved that the sequence produced by $\rho(F_2(P_n))$ coincides with the sequence $A085680$ in OEIS [18], i.e., they proved Pratt's conjecture. So, the problem of determining $\rho(F_3(P_n))$ arises naturally. As in [19], it would be interesting to relate the problem of determining $\rho(F_3(P_n))$ to finding the largest S' of length n and constant weight $w = 3$.

In this paper, we deal with the problem of determining $\rho(F_3(P_n))$. The rest of the paper is organized as follows: In Section 2, we give the definition of some concepts and prove some propositions which will be useful throughout the rest of the paper. In Section 3, we prove that Γ_n^k and $F_k(P_n)$ are isomorphic when $w = k$. It is easy to see that, if $\Gamma_n^k \simeq F_k(P_n)$, then $\rho(\Gamma_n^k) = \rho(F_k(P_n))$ is the maximum cardinality of a binary code with constant weight k that can correct a single adjacent transposition. Since the maximum packing set problem is computationally equivalent to the maximum independent set problem, which is an NP-hard problem [8, 11], then no polynomial time algorithms are expected to be found. Nevertheless, we have developed an exact algorithm for some instances in Section 4. That is, we compute the exact value of $\rho(F_3(P_n))$ for $n \leq 12$. In Section 5, using the distance matrix (DM) of a graph and improving the constructions proposed in [15], we present some algorithms that give a lower bound for $\rho(F_3(P_n))$ with $13 \leq n \leq 44$. We remark that our lower bound for $\rho(F_3(P_n))$ is better than those in [15]. Finally, in Section 6 we give an upper bound for $\rho(F_3(P_n))$ with $n \geq 13$.

2. Definitions and preliminaries

Let G be a graph with vertex and edge sets $V(G) = \{v_1, v_2, \ldots, v_n\}$ and $E(G)$, respectively.

1) Let G' be a graph, with vertex and edge sets $V(G')$ and $E(G')$, respectively, such that $V(G') \subseteq V(G)$ and $E(G') \subseteq E(G)$. Then, G' is a *subgraph* of G (and G is a *supergraph* of G') and we write

$G' \subseteq G$. Now, if $G' \subseteq G$ and G' contains all the edges $xy \in E(G)$ such that $x, y \in V(G')$, then G' is an *induced subgraph* of G. If $S \subseteq V(G)$, then $G[S]$ is a subgraph of G induced by S.

2) *The complement* G^c of a graph G is the graph with vertex set $V(G)$ such that two distinct vertices of G^c are adjacent if and only if they are not adjacent in G.
3) The *neighborhood* of a vertex $v \in V(G)$ is $N_G(v) := \{u \in V : uv \in E(G)\}$, and given a set $S \subset V(G)$ we define $N_G(S) := \bigcup_{v \in S} N_G(v)$.
4) Let S be a subset of $V(G)$. Then, S is called an *independent set* of G if no two vertices of S are adjacent in G, and the *independence number* $\alpha(G)$ of G is the maximum cardinality of an independent set of G, that is $\alpha(G) := \max_{S \subseteq V(G)} \{|S| \ : \ S \text{ is an independent set}\}$
5) Let $u, v \in V(G)$. The *distance* between u and v in G, denoted by $d_G(u, v)$, is the length of the shortest path between u and v. Let k be a positive integer. A set $T \subseteq V(G)$ is a *k-packing set* of G if every pair of distinct vertices $u, v \in T$ satisfy $d_G(u, v) \geq k + 1$. The *packing number* $\rho(G)$ of G is the maximum cardinality of a packing set of G, that is, $\rho(G) := \max_{T \subseteq V(G)} \{|T| \ : \ T \text{ is a packing set}\}$.
   If $k = 2$, then T will be a *2-packing set* (or simply, *a packing set*) of G. Moreover, it is easy to see that an independent set of G is also a 1-*packing set* of G.
6) Let G be a graph and $F \subseteq E(G^c)$. We define $G + F$ as the graph whose vertex and edge sets are as follows: $V(G + F) = V(G)$ and $E(G + F) = E(G) \cup F$. If $F = \{a, b\}$, then we will only write $G + ab$ instead of $G + F$.
7) Let G_1 and G_2 be two graphs. We will say that G_1 and G_2 are *isomorphic* if there is a bijection $f : V(G_1) \to V(G_2)$ such that $uv \in E(G_1)$ if and only if $f(u)f(v) \in E(G_2)$ for all $u, v \in V(G_1)$. If G_1 and G_2 are isomorphic, then we write $G_1 \simeq G_2$ and the map f is *an isomorphism*.
8) Let a_{ij} be the shortest path length between v_i and v_j in G. The *distance matrix* of G, denoted by $DM(G)$, is an $n \times n$ matrix whose $(i, j)^{th}$ entry is a_{ij}. Clearly, $DM(G)$ is a symmetric matrix with trace equal to zero.

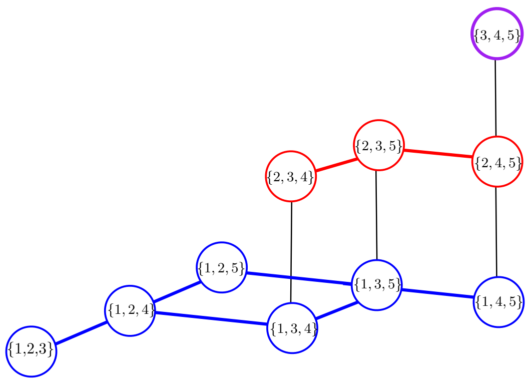

Figure 2. The 3-Token graph $F_3(P_5)$ of P_5. We remark that $F_3(P_5)$ can be drawn as a pyramid with 3 floors.

For instance, the distance matrix $DM(F_3(P_5))$ of $F_3(P_5)$ (see Figure 2), is the square matrix of size $\binom{5}{2} \times \binom{5}{2}$ as depicted in Figure 3, where $v_0 = \{1, 2, 3\}, v_1 = \{1, 2, 4\}, v_2 = \{1, 2, 5\}, v_3 = \{1, 3, 4\}, v_4 = \{1, 3, 5\}, v_5 = \{2, 3, 4\}, v_6 = \{1, 4, 5\}, v_7 = \{2, 3, 5\}, v_8 = \{2, 4, 5\}, v_9 = \{3, 4, 5\}$, according to *Algorithm 1*.

Algorithm 1: Algorithm to construct $F_3(P_n)$ and $(F_3(P_n))^2$ with $n \geq 3$.

Input: Graph P_n, with $n \geq 3$.

Output: $F_3(P_n)$, $(F_3(P_n))^2$ with $n \geq 3$ and ID assignment to each vertex of $F_3(P_n)$.

```
Compute the (n choose 3) 3-subsets of V(P_n) and store them in lexicographical order in a vector (namely, vector);
F_3Set ← {};
sim_dif ← {};
for i ← 0 to vector.size() − 1 do
    for j ← i + 1 to vector.size() − 1 do
        sim_dif ← vector[i]△vector[j] ;
        cardinality ← sim_dif.size() ;
        if cardinality == 2 then
            if sim_dif ∈ E(P_n) then
                if vector[i] and vector[j] do not have ID then
                    F_3Set.push_back(vector[i]);
                    vector[i].ID ← F_3Set.size() − 1;
                    F_3Set.push_back(vector[j]);
                    vector[j].ID ← F_3Set.size() − 1;
                    Add an edge between vector[i] and vector[j] in F_3(P_n);
                end
                else if vector[i] has an ID and vector[j] has no ID then
                    vector[i] keeps its ID;
                    F_3Set.push_back(vector[j]);
                    vector[j].ID ← F_3Set.size() − 1;
                    Add an edge between vector[i] and vector[j] in F_3(P_n);
                end
                else if vector[i] and vector[j] have ID then
                    vector[i] keeps its ID;
                    vector[j] keeps its ID;
                    Add an edge between vector[i] and vector[j] in F_3(P_n);
                end
            end
            else if sim_dif ∉ E(P_n) then
                vector[i] keeps its ID;
                Do not add edge between vector[i] and vector[j] in F_3(P_n);
            end
        end
        else if cardinality ≠ 2 then
            vector[i] keeps its ID;
            Do not add edge between vector[i] and vector[j] in F_3(P_n);
        end
    end
end
(F_3(P_n))^2 ← power(F_3(P_n), 2)
```

	v_0	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9
v_0	0	1	2	2	3	3	4	4	5	6
v_1	1	0	1	1	2	2	3	3	4	5
v_2	2	1	0	2	1	3	2	2	3	4
v_3	2	1	2	0	1	1	2	2	3	4
v_4	3	2	1	1	0	2	1	1	2	3
v_5	3	2	3	1	2	0	3	1	2	3
v_6	4	3	2	2	1	3	0	2	1	2
v_7	4	3	2	2	1	1	2	0	1	2
v_8	5	4	3	3	2	2	1	1	0	1
v_9	6	5	4	4	3	3	2	2	1	0

Figure 3. The distance matrix $DM(F_3(P_5))$ of $F_3(P_5)$.

The *k-th power* of G, denoted by G^k, is the graph with vertex set $V(G^k) = V(G)$ such that two vertices u, v are adjacent in G^k if and only if $d_G(u, v) \leq k$. Then, G^2 has vertex set $V(G)$ and its edges are given by the following:

$$\begin{cases} ab \in E(G) \rightarrow ab \in E(G^2), \\ \text{if } ab \notin E(G) \text{ and } d_G(a, b) = 2, \text{ then } ab \in E(G^2), \text{ for } a, b \in V(G). \end{cases}$$

From the involved definitions, we have the next results.

Proposition 1. *$S \subseteq V(G)$ is a packing set of G if and only if S is an independent set of G^2.*

Proof. Let S be a packing set of G. Then, for every pair of distinct vertices $u, v \in S$, it follows that $d_G(u, v) \geq 3$, in particular $u \notin N_{G^2}(v)$ and $v \notin N_{G^2}(u)$. Hence, S is an independent set of G^2. On the other hand, suppose that S is an independent set of G^2. Seeking a contradiction, suppose that S is not a packing set of G. Then, there are at least two vertices $u, v \in S$ such that $d_G(u, v) \leq 2$. Hence, $v \in N_{G^2}(u)$ and $u \in N_{G^2}(v)$, which contradicts that S is an independent set of G^2.

The next corollary is a consequence of Proposition 1.

Corollary 1. $\rho(G) = \alpha(G^2)$.

The next proposition will be useful.

Proposition 2. *Let G be a graph and let u, v be two vertices of $V(G)$ such that $uv \notin E(G)$. Then, $\alpha(G + uv) \leq \alpha(G)$.*

Proof. Let S be an independent set of G with maximum cardinality, i.e., $\alpha(G) = |S|$. Let $u, v \in V(G)$ such that $uv \notin E(G)$. Let S' be an independent set with maximum cardinality of $G + uv$. First, suppose that $u \notin S$ or $v \notin S$. We deal with the case when $u \notin S$. Note that the case $v \notin S$ can be handled in a similar way. Adding the edge uv to G, we have $S' = S$ and so $\alpha(G + uv) = \alpha(G)$. Now, we may assume that $u, v \in S$. If we add the edge uv to G, then $u \in N_{G+uv}(v)$ and $v \in N_{G+uv}(u)$. So, $S' = S \setminus \{w\}$ with $w \in \{u, v\}$. Hence, $\alpha(G') = |S'| \leq \alpha(G) = |S|$, as desired.

Corollary 2. *Let G be a graph and let $A \subseteq V(G)$. Let $E^*(G^c)$ be the subset of $E(G^c)$ such that $E^*(G^c) := \{uv \in E(G^c) \mid u, v \in A\}$. Then, $\alpha(G + E^*(G^c)) \leq \alpha(G)$.*

3. Γ_n^k and $F_k(P_n)$ are isomorphic

In the next proposition, we prove that Γ_n^k and $F_k(P_n)$ are isomorphic. Hence, $\rho(\Gamma_n^k) = \rho(F_k(P_n))$ is the maximum cardinality of a binary code of length n with constant weight k that can correct a single adjacent transposition.

Proposition 3. *Let $n \geq 3$ and $k \leq n-1$ be two positive integers. Let P_n be a path graph with n vertices and let $F_k(P_n)$ be its k-token graph. Let Γ_n^k be a binary code graph of length $n \geq 3$ and constant weight $w = k$. Then, $\Gamma_n^k \simeq F_k(P_n)$.*

Proof. Let $V(P_n) = \{1, \ldots, n\}$ and $E(P_n) = \{\{i, i+1\} : 1 \leq i \leq n-1\}$ be, respectively, the vertex and edge sets of P_n. Let ψ be a map defined as follows:

$$\psi\colon V(F_k(P_n)) \longrightarrow V(\Gamma_n^k)$$
$$B \longmapsto (b_1, b_2, \ldots, b_n), \text{ with } b_i = \begin{cases} 1, & \text{if } i \in B; \\ 0, & \text{otherwise.} \end{cases}$$

We prove that ψ is bijective. Since P_n is a finite graph of order n, then $|V(F_k(P_n))| = \binom{n}{k}$ [6]. On the other hand, from the definition of Γ_n^k, it follows that $|V(F_k(P_n))| = |V(\Gamma_n^k)|$. Then, it is enough to show that ψ is injective. Let A and B be two $k-$subsets of $V(F_k(P_n))$ such that $\psi(A) = (a_1, a_2, \ldots, a_n)$ and $\psi(B) = (b_1, b_2, \ldots, b_n)$. Assume that $\psi(A) = \psi(B)$, then $a_i = b_i$ for all $i \in \{1, 2, \ldots, n\}$. Since $a_i = b_i = \begin{cases} 1, & \text{if } i \in A; \\ 0, & \text{otherwise.} \end{cases}$ Then, $A = B$. Hence, ψ is injective.

On the other hand, let A and B be two adjacent vertices of $F_k(P_n)$. Since A and B are two k-subsets of $\{1, \ldots, n\}$, then there is $j \in \{1, \ldots, n-1\}$ such that $A \triangle B = \{j, j+1\} \in E(P_n)$. Without loss of generality, we assume that $j \in A$, and then $(j+1) \in B$. Clearly, $(j+1) \notin A$ and $j \notin B$. Then,

$$\psi(A) = (a_1, a_2, \ldots, a_{j-1}, \overbrace{1}^{j\text{–th bit,}}, \overbrace{0}^{(j+1)\text{–th bit}}, a_{j+2}, \ldots, a_n),$$

and

$$\psi(B) = (a_1, a_2, \ldots, a_{j-1}, \overbrace{0}^{j\text{–th bit,}}, \overbrace{1}^{(j+1)\text{–th bit}}, a_{j+2}, \ldots, a_n).$$

Since $\psi(B)$ is obtained from $\psi(A)$ by transposing contiguous bits, then $\psi(A)$ and $\psi(B)$ are adjacent in Γ_n^k. Conversely, it is easy to check that if $\psi(A)$ and $\psi(B)$ are adjacent in Γ_n^k, then A and B are adjacent in $F_k(P_n)$, as desired.

Corollary 3. *Let P_n be a path graph with $n \geq 3$ vertices and let $F_3(P_n)$ be its $3-$token graph. Let Γ_n^3 be a binary code graph of length $n \geq 3$ and constant weight 3. Then, $\Gamma_n^3 \simeq F_3(P_n)$.*

From Corollary 3, it follows that $\rho(\Gamma_n^3) = \rho(F_3(P_n))$.

4. Exact value of $\rho(F_3(P_n))$ for $n \leq 12$

In this section, we give an algorithm that computes the exact value of $\rho(F_3(P_n))$ for $n \leq 12$. We have used Corollary 1 and the fact that there is a function that Mathematica (Wolfram Language) has available to determine the independence number of graphs [20].

Algorithm 1 is used to construct $F_3(P_n)$, $(F_3(P_n))^2$ and assigns an identification (ID) to each vertex of $F_3(P_n)$. Furthermore, the same algorithm sorts the vertices of $F_3(P_n)$ according to their respective ID and stores them in the set F_3Set. This order is based on the lexicographic order and the adjacency of the vertices in $F_3(P_n)$. See *Comments on Algorithm 1* for additional details. On the other hand, using the set F_3Set, we construct the distance matrix of $F_3(P_n)$, which is important in *Algorithm 2*. Since F_3Set is unique for a given $F_3(P_n)$, then the distance matrix of $F_3(P_n)$ is also unique. See $DM(F_3(P_5))$ in Figure 3. We sometimes use the index of an element in F_3Set to refer to it. For example, in *Algorithm 2* we use i to refer to v_i.

Algorithm 2: Algorithm to determine a lower bound for $\rho(F_3(P_n))$ with $n \geq 13$.

Input: Graph P_n, with $n \geq 13$.

Output: Compute a lower bound for $\rho(F_3(P_n))$ with $n \geq 13$.

```
while n ≥ 13 do
    Construct F3(Pn) and the vertex set F3Set by using Algorithm 1;
    DM(F3(Pn)) ← (a_ij)_ij ;
    packing_max ← 0;
    for i ← 0 to F3Set.size() − 1 do
        probable_packing ← {} ;
        for j ← 0 to F3Set.size() − 1 do
            if a_ij == 0 then
                probable_packing.insert(j) ;
            end
            else if a_ij ≥ 3 then
                probable_packing.push_back(j)
            end
        end
        for i ← 1 to probable_packing.size() − 2 do
            for j ← i + 1 to probable_packing.size() − 1 do
                if a_ij < 3 then
                    probable_packing.erase(j);
                    j ← j − 1;
                end
            end
        end
        packing_max ← max(packing_max, probable_packing.size());
    end
end
```

4.1. *Comments on* Algorithm 1

Algorithm 1 has been developed in Python [16], and involves the following questions: (*i*) how is the ID assigned to each vertex of $V(F_3(P_n))$, that is, how are the nodes of $V(F_3(P_n))$ ordered in the set F_3Set? and (*ii*) how is the graph $F_3(P_n)$ constructed? Let us explain how *Algorithm 1* works using an example. Consider the path P_5 as input, then the expected output is $F_3(P_5)$, $(F_3(P_5))^2$, and ID assignment to each vertex of $F_3(P_5)$ as in Figure 2. First, the $\binom{5}{3}$ 3-subsets of $V(P_5)$ are computed and stored in lexicographical order in a vector as:

$$vector := \{\{1,2,3\},\{1,2,4\},\{1,2,5\},\{1,3,4\},\{1,3,5\},\{1,4,5\},\{2,3,4\},\{2,3,5\},\{2,4,5\},\{3,4,5\}\}$$

Next, we assign the ID to each vertex of $V(F_3(P_5))$ and we construct F_3Set and $F_3(P_5)$ as follows:

A_1) If $i = 0$ and $j = 1$, then $\{1,2,3\}\triangle\{1,2,4\} = \{3,4\} \in E(P_5)$. Clearly, $\{1,2,3\}$ and $\{1,2,4\}$ do not have ID, then $F_3Set = \{\{1,2,3\}\}$ and $\{1,2,3\}$ becomes the node with ID 0, next, $F_3Set = \{\{1,2,3\},\{1,2,4\}\}$ and $\{1,2,4\}$ with ID 1. Moreover, $\{1,2,3\}$ and $\{1,2,4\}$ are adjacent in $F_3(P_5)$. Hence, $F_3Set = \{\{1,2,3\},\{1,2,4\}\}$.

A_2) If $i = 0$ and $j = 2$, then $\{1,2,3\}\triangle\{1,2,5\} = \{3,5\} \notin P_5$. Then, $\{1,2,3\}$ stays as the node with ID 0 and $\{1,2,5\}$ is not added as a vertex of F_3Set. $\{1,2,3\}$ and $\{1,2,5\}$ are not adjacent in $F_3(P_5)$. So, $F_3Set = \{\{1,2,3\},\{1,2,4\}\}$.

A_3) If $i = 0$ and $j = 3$, then $\{1,2,3\}\triangle\{1,3,4\} = \{2,4\} \notin P_5$. Then, $\{1,2,3\}$ remains as the node with ID 0 and $\{1,3,4\}$ does not belong to F_3Set. $\{1,2,3\}$ and $\{1,3,4\}$ are not adjacent in $F_3(P_5)$. So, $F_3Set = \{\{1,2,3\},\{1,2,4\}\}$.

A_4) If $i = 0$ and $j = 4$, then $\{1,2,3\}\triangle\{1,3,5\} = \{2,5\} \notin P_5$. Then, $\{1,2,3\}$ remains as the node with ID 0 and $\{1,3,5\}$ is not added as a vertex of F_3Set. $\{1,2,3\}$ and $\{1,3,5\}$ are not adjacent in $F_3(P_5)$. So, $F_3Set = \{\{1,2,3\},\{1,2,4\}\}$.

A_5) If $i = 0$ and $j = 5$, then $\{1,2,3\}\triangle\{1,4,5\} = \{2,3,4,5\}$, so $|\{1,2,3\}\triangle\{1,4,5\}| \neq 2$. Then $\{1,2,3\}$ remains as the node with ID 0 and $\{1,3,4\}$ is not added as a vertex of F_3Set. $\{1,2,3\}$ and $\{1,4,5\}$ are not adjacent in $F_3(P_5)$. So, $F_3Set = \{\{1,2,3\},\{1,2,4\}\}$.

Continuing with this procedure until j reaches the upper limit of the loop, it is easy to see that $F_3Set = \{\{1,2,3\},\{1,2,4\}\}$.

B_1) If $i = 1$ and $j = 2$, then $\{1,2,4\}\triangle\{1,2,5\} = \{4,5\} \in E(P_5)$. Clearly, $\{1,2,4\}$ has an ID, but $\{1,2,5\}$ has no ID. Then, $\{1,2,4\}$ stays with ID 1, also $F_3Set = \{\{1,2,3\},\{1,2,4\},\{1,2,5\}\}$ and $\{1,2,5\}$ has ID 2. $\{1,2,4\}$ and $\{1,2,5\}$ are adjacent in $F_3(P_5)$. So, $F_3Set = \{\{1,2,3\},\{1,2,4\},\{1,2,5\}\}$.

B_2) If $i = 1$ and $j = 3$, then $\{1,2,4\}\triangle\{1,3,4\} = \{2,3\} \in E(P_5)$. Clearly, $\{1,2,4\}$ has an ID, but $\{1,3,4\}$ has no ID. Then , $\{1,2,4\}$ keeps its ID 1, also $F_3Set = \{\{1,2,3\},\{1,2,4\},\{1,2,5\},\{1,3,4\}\}$ and $\{1,3,4\}$ becomes the node with ID 3. $\{1,2,4\}$ and $\{1,3,4\}$ are adjacent in $F_3(P_5)$. Hence, $F_3Set = \{\{1,2,3\},\{1,2,4\},\{1,2,5\},\{1,3,4\}\}$.

Again, continuing with this procedure until j reaches the upper limit of the loop, it follows that $F_3Set = \{\{1,2,3\},\{1,2,4\},\{1,2,5\},\{1,3,4\}\}$.

The algorithm continues until all the 3-subsets of $V(P_5)$ are pairwise compared, obtaining

$$F_3Set = \{\{1,2,3\},\{1,2,4\},\{1,2,5\},\{1,3,4\},\{1,3,5\},\{2,3,4\},\{1,4,5\},\{2,3,5\},\{2,4,5\},\{3,4,5\}\}. \tag{4.1}$$

Finally, $F_3(P_n)$ and also $(F_3(P_5))^2$ are constructed.

It follows that $F_3Set = \{v_0, v_1, \ldots, v_9\}$ with

$$v_0 = \{1,2,3\}, v_1 = \{1,2,4\}, v_2 = \{1,2,5\}, v_3 = \{1,3,4\}, v_4 = \{1,3,5\},$$
$$v_5 = \{2,3,4\}, v_6 = \{1,4,5\}, v_7 = \{2,3,5\}, v_8 = \{2,4,5\}, v_9 = \{3,4,5\} \quad (4.2)$$

The graphs $F_3(P_5)$ and $(F_3(P_5))^2$ are depicted in Figure 4.

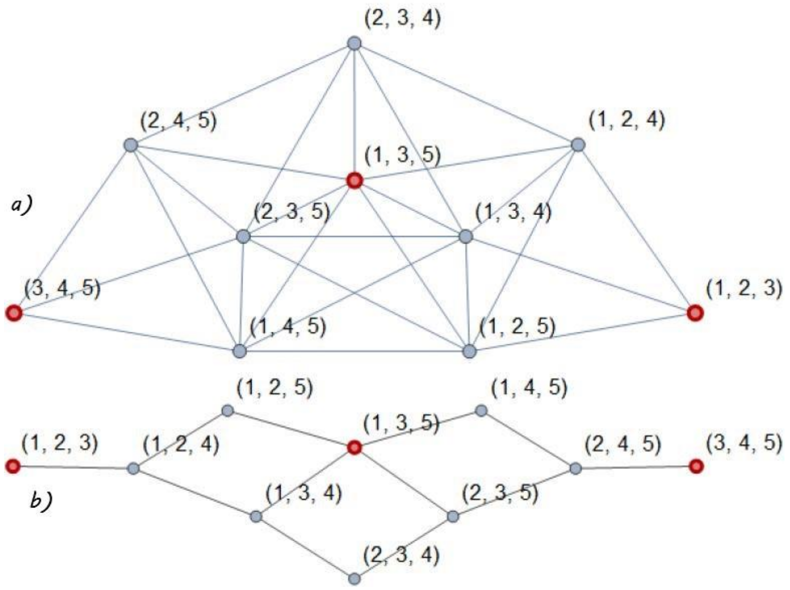

Figure 4. The graph in *a)* is $(F_3(P_5))^2$. It is obtained by adding all the edges uv such that $d(u, v) = 2$ to $F_3(P_5)$. The red vertices in $(F_3(P_5))^2$ form a maximum independent set and so then $\alpha((F_3(P_5))^2) = 3$. In *b)* it is shown the corresponding packing set of $F_3(P_5)$.

With *Algorithm 3* implemented in Mathematica (Wolfram Language), we have computed the exact value of the packing number of $F_3(P_n)$ for $n \leq 12$.

Algorithm 3: Algorithm to determine the exact value of the packing number of $F_3(P_n)$, for $n \leq 12$.

Input: Graph P_n, with $n \geq 3$.
Output: Compute $\rho(F_3(P_n))$ for $n \geq 3$.

```
while n ≥ 3 do
    Construct (F_3(P_n))^2 by using Algorithm 1;
    G ← (F_3(P_n))^2;
    ρ(F_3(P_n)) ← α(G)
end
```

With *Algorithm 3* we find a maximum independent set of $(F_3(P_n))^2$. Then, we find $\rho(F_3(P_n))$ by using Corollary 1 in Figure 4 is depicted $\rho(F_3(P_5))$.

From Corollary 3 and *Algorithm 3*, the size of the largest binary code S' of length $n \leq 12$ and constant weight 3 is given in Table 1.

Table 1. The exact value of $\rho(F_3(P_n))$ for $n \in \{3, \ldots, 12\}$.

n	3	4	5	6	7	8	9	10	11	12
$\rho(F_3(P_n)) = \rho(\Gamma_n^3)$	1	2	3	6	9	13	18	24	32	41

For $n = 12$, we have $|V(F_3(P_n))| = |V((F_3(P_n))^2)| = 220$. Although we obtained $\rho(F_3(P_n)) = 41$, it is important to note that the CPU time required was a bit long. On the other hand, when $n \geq 13$, we have $|V(F_3(P_n))| = |V((F_3(P_n))^2)| \geq 286$, and the graph $(F_3(P_n))^2$ starts to be dense. Unfortunately, the processing time required is too long when we use *Algorithm 3*.

5. A lower bound for $\rho(F_3(P_n))$

In this section we deal with another algorithm (namely, *Algorithm 2*) for computing a lower bound for $\rho(F_3(P_n))$. In Table 3 [9], we summarize the improved lower bounds for $\rho(F_3(P_n))$ with $13 \leq n \leq 44$.

5.1. Comments on Algorithm 2

We construct $F_3(P_n)$ and the vertex set F_3Set by using *Algorithm 1*. Then we use a function that Mathematica (Wolfram Language) has available to obtain the distance matrix $DM(F_3(P_n))$ of size $F_3Set.size() \times F_3Set().size()$. And, to find the $\rho(F_3(P_n))$, we have developed software in C++.

The variable *packing_max* is used to store the packing number found. Additionally, the set *probable_packing* stores the possible packing nodes. For each node $v_i \in F_3Set$, we check in the $DM(F_3(P_n))$ the distance between v_i and any other node $v_j \in F_3Set$.

If a_{ij} is zero, i.e., $i = j$, then we store the current node v_j in the first position of *probable_packing* using *probable_packing.insert(j)*.

Furthermore, the rest of the nodes v_j with $j \neq i$, which are at a distance 3 from v_i, are stored after the node v_j with $j = i$ in *probable_packing* using *probable_packing.push_back(j)*.

Once we have the probable packing set, with the next **For loops**, we ensure that all nodes in the probable packing set are kept at a distance of at least 3 from each other.

By taking each vertex $v_j \in F_3Set$ such that $j = i$ as the first element of *probable_packing*, we have obtained some good results when one of the nodes v_0, v_1, and v_2 in F_3Set is the second element of *probable_packing*. In Table 2, we present some results for $13 \leq n \leq 32$.

Table 2. Some lower bounds for $\rho(F_3(P_n))$ with $13 \leq n \leq 32$.

n	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
Our results with v_0 as second node	50	61	75	90	106	127	146	170	198	226	255	291	327	366	413	454	500	563	612	667
Our results with v_1 as second node	49	61	73	90	108	125	149	170	196	226	257	290	326	367	408	455	504	558	612	679
Our results with v_2 as second node	49	61	75	91	109	126	146	171	197	223	257	291	324	366	410	451	502	559	607	672
Maximum value	50	61	75	91	109	127	149	171	198	226	257	291	327	367	413	455	504	563	612	679

5.2. Improving the lower bounds of $\rho(F_3(P_n))$ for $n \in \{14, 17, 20, 23, 26\}$

For $13 \leq n \leq 26$, we observe that the constructions proposed in [15] of finding $\rho(F_3(P_n))$ when $n \equiv 2 \pmod 3$ can be improved.

We will now explain how we get such results. As in [15], we consider the path P_n with $\{j, j+1\} \in E(P_n)$ for $1 \leq j < n$. Without loss of generality, we will write the elements of each vertex $\{i_1, i_2, i_3\} \in V(F_3(P_n))$ in ascending order, i.e., we will assume that $i_1 < i_2 < i_3$.

Let $n \geq 3$ be an integer and let $t \in \{0, 2\}$. We define the sets of vertices $B(n, t)$ and $P(n)$ as follows:

$$B(n,t) := \begin{cases} \bigcup_{j=0}^{(n-3)/3}(\bigcup_{k=j+1}^{n/3}\{\{1, 3j+2, 3k\}\}) & \text{if } n \equiv 0 \pmod 3, \\ \bigcup_{j=0}^{(n-4)/3}(\bigcup_{k=j+1}^{(n-1)/3}\{\{1, 3j+2, 3k\}\}) & \text{if } n \equiv 1 \pmod 3, \\ \bigcup_{j=0}^{(n-5)/3}(\bigcup_{k=j+1}^{(n-2)/3}\{\{1, 3j+2, 3k+t\}\}) & \text{if } n \equiv 2 \pmod 3. \end{cases} \tag{5.1}$$

Clearly, if $t = 0$, we have the lower bounds given in [15]. Now, suppose that $t = 2$. Then, $P(n) := (B(n,2) \bigcup_{k=0}^{(n-8)/3}\{\{1, 4+3k, 6+3k\}\}) \bigcup_{k=1}^{n-2} B(n-k, 0)$ is a packing set of $F_3(P_n)$. Indeed, the set $\bigcup_{k=0}^{n-2} B(n-k, 0)$ is a packing set of $F_3(P_n)$, see [15], and $B(n, 2)$ is a slight refinement to the packing proposed in [15]. Note that, if $n \geq 8$ and $n \equiv 2 \pmod 3$, then $B(n, 2)$ and the set $\bigcup_{k=0}^{n-2} B(n-k, 0)$ allows us to add the vertices $\bigcup_{k=0}^{(n-8)/3}\{\{1, 4+3k, 6+3k\}\}$ to the packing set. It is easy to see that,

i) for each pair of vertices x, y in $B(n, 2) \bigcup_{k=0}^{(n-8)/3}\{\{1, 4+3k, 6+3k\}\}$ we have $d_{F_3(P_n)}(x, y) \geq 3$ and,
ii) if $n \geq 8$ and $n \equiv 2 \pmod 3$, then for each pair of vertices x, y in $P(n)$ we have $d_{F_3(P_n)}(x, y) \geq 3$.

Therefore, $P(n)$ is a packing set of $F_3(P_n)$ whenever $n \geq 14$ and $n \equiv 2 \pmod 3$.

As an example, let us take $n = 14$. Then,

$$B(14, 2) = \bigcup_{j=0}^{3}(\bigcup_{k=j+1}^{4}\{\{1, 3j+2, 3k+2\}\}) =$$
$$= \{\{1,2,5\}, \{1,2,8\}, \{1,2,11\}, \{1,2,14\},$$
$$\{1,5,8\}, \{1,5,11\}, \{1,5,14\}, \{1,8,11\}, \{1,8,14\}, \{1,11,14\}\},$$

$\bigcup_{k=0}^{(14-8)/3}\{\{1, 4+3k, 6+3k\}\} = \{\{1,4,6\}, \{1,7,9\}, \{1,10,12\}\}$,
and

$$\bigcup_{k=1}^{12} B(14-k, 0) = \{\{2,3,4\}, \{2,3,7\}, \{2,3,10\},$$
$$\{2,3,13\}, \{2,6,7\}, \{2,6,10\}, \{2,6,13\}, \{2,9,10\}, \{2,9,13\},$$
$$\{2,12,13\}, \{3,4,5\}, \{3,4,8\}, \{3,4,11\}, \{3,4,14\}, \{3,7,8\},$$
$$\{3,7,11\}, \{3,7,14\}, \{3,10,11\}, \{3,10,14\}, \{3,13,14\}$$
$$, \{4,5,6\}, \{4,5,9\}, \{4,5,12\}, \{4,8,9\}, \{4,8,12\}, \{4,11,12\},$$
$$\{5,6,7\}, \{5,6,10\}, \{5,6,13\}, \{5,9,10\}, \{5,9,13\},$$
$$\{5,12,13\}, \{6,7,8\}, \{6,7,11\}, \{6,7,14\}, \{6,10,11\}, \{6,10,14\},$$
$$\{6,13,14\}, \{9,10,14\}, \{9,13,14\}, \{10,11,12\},$$
$$\{7,8,9\}, \{7,8,12\}, \{7,11,12\}, \{8,9,10\}, \{8,9,13\}, \{8,12,13\},$$

$\{9, 10, 11\}, \{11, 12, 13\}, \{12, 13, 14\}\}$.

The set $P(14)$ is a packing set of $F_3(P_{14})$ with $|P(14)| = |B(14, 2)| + |\bigcup_{k=0}^{(14-8)/3}\{\{1, 4 + 3k, 6 + 3k\}\}| + |\bigcup_{k=1}^{12} B(14 - k, 0)| = 10 + 3 + 50 = 63$. By using this procedure, we obtained $|P(17)| = 109$, $|P(20)| = 173$, $|P(23)| = 258$ and $|P(26)| = 367$.

Table 3 contains some lower bounds for $\rho(F_3(P_n))$ with $13 \leq n \leq 32$. The interested reader can find in Table 3 [9] the lower bounds for $\rho(F_3(P_n))$ with $13 \leq n \leq 44$. We remark that our results are better than those presented in [15].

Table 3. Our lower bounds for $\rho(F_3(P_n))$ with $13 \leq n \leq 32$.

n	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
Results in [15]	50	60	75	90	105	126	147	168	196	224	252	288	324	360	405	450	495	550	605	660
Our results	50	63	75	91	109	127	149	173	198	226	258	291	327	367	413	455	504	563	612	679

6. An upper bound for $\rho(F_3(P_n))$ with $n \geq 13$

Recently, Alba et al. [4] established the following statements for the independence number of $\rho(F_3(P_n))$.

Definition 1. *(Definition 3.1 [4]) Let G be a bipartite graph with bipartition $\{A, B\}$. Let $m := |A| \geq 1, n := |B| \geq 1$, and let $k \in \{1, \ldots, m + n - 1\}$. Let $\mathcal{A} := \{K \subset V(G) : |K| = k, |A \cap K|$ is odd$\}$, and let $\mathcal{B} := \{K \subset V(G) : |K| = k, |A \cap K|$ is even$\}$.*

From Definition 1 and Proposition 12 in [6] we know that $F_3(P_n)$ is a bipartite graph with bipartition $\{\mathcal{A}, \mathcal{B}\}$, where $\mathcal{A} := \{K \subset V(G) : |K| = 3, |A \cap K|$ is odd$\}$ and $\mathcal{B} := \{K \subset V(G) : |K| = 3, |A \cap K|$ is even$\}$.

Theorem 1. *(Theorem 3.9 [4]) If G' is a bipartite supergraph of G with bipartition $\{A, B\}$, and G has either a perfect matching or an almost perfect matching, then $\alpha(F_k(G')) = \max\{|\mathcal{A}|, |\mathcal{B}|\}$.*

Since $G = P_n$ satisfies the hypotheses of Theorem 1, we can conclude that $\alpha(F_3(P_n)) = \max\{|\mathcal{A}|, |\mathcal{B}|\}$.

Corollary 4. *(Corollary 3.10 [4]) Let $t \in \mathbb{Z}^+$. If $G \in \{P_t, C_{2t}, K_{t,t+1}\}$ and k is an integer such that $1 \leq k \leq |G| - 1$, then $\alpha(F_k(G)) = \max\{r, \binom{p}{k} - r\}$, where $p := |G| = t$ and $r := \sum_{i=1}^{\lceil k/2 \rceil} \binom{\lceil p/2 \rceil}{2i-1}\binom{\lfloor p/2 \rfloor}{k-2i+1}$.*

The following proposition is an easy consequence of Corollary 4.

Proposition 4. *Let n and m be two positive integers. Then,*

$$\alpha(F_3(P_n)) = \begin{cases} \binom{2m}{3}/2, & \text{if } n = 2m, \\ \frac{m}{3}(2m^2 + 1), & \text{if } n = 2m + 1. \end{cases}$$

In view of Corollaries 1 and 2 and Proposition 4, we have the following Proposition 5.

Proposition 5. *Let n and m be two positive integers such that $m \geq 8$. Then,*

$$\rho(F_3(P_n)) \leq \begin{cases} \lceil \binom{2m}{3}/6 \rceil, & \text{if } n = 2m, \\ \lceil \frac{m}{9}(2m^2 + 1) \rceil, & \text{if } n = 2m + 1. \end{cases}$$

Proof. Let n and m be as in the statement. By Corollary 1, it follows that $\rho(F_3(P_n)) = \alpha((F_3(P_n))^2)$. Now, let $E^*((F_3(P_n))^c) := \{uv \in E((F_3(P_n))^c) \mid d_{F_3(P_n)}(u,v) = 2 \text{ with } u,v \in V(F_3(P_n))\}$. From the definition of $(F_3(P_n))^2$ it is easy to see that $(F_3(P_n))^2 = F_3(P_n) + E^*((F_3(P_n))^c)$. On the other hand, Corollary 2 and Proposition 4 imply that $\alpha((F_3(P_n))^2) = \alpha(F_3(P_n) + E^*((F_3(P_n))^c)) \leq \alpha(F_3(P_n)) = \binom{n}{3}/2$. Now, since $F_3(P_n)$ is a bipartite graph, then we have a partition of $V(F_3(P_n)) = \{\mathcal{A}, \mathcal{B}\}$. Note that either $\mathcal{A}$ or $\mathcal{B}$ is an independent set of $F_3(P_n)$ of maximum cardinality. Without loss of generality, we suppose that $\mathcal{A}$ is an independent set of $F_3(P_n)$ of maximum cardinality. From the definition of $F_3(P_n)$ with $n \geq 16$, it is easy to see that only the vertices $v_0 := \{1,2,3\}$ and $v_{\binom{n}{3}-1} := \{(n-2),(n-1),n\}$ have degree 1. Moreover, we note that v_0 (respectively, $v_{\binom{n}{3}-1}$) is at distance two from $v_2 := \{1,2,5\}$ and $v_3 := \{1,3,4\}$ (respectively, $v_{\binom{n}{3}-4} := \{(n-4),(n-1),n\}$ and $v_{\binom{n}{3}-3} := \{(n-3),(n-2),n\}$). Also, it is easy to see that v_2 is at distance two from v_3, and $v_{\binom{n}{3}-3}$ is at distance two from $v_{\binom{n}{3}-4}$. Similarly, we observe that each vertex in $V(F_3(P_n))$ is at distance two from at least two vertices (namely, u and w) of $V(F_3(P_n))$ such that $d_{F_3(P_n)}(u,w) = 2$. Then, for each $v \in \mathcal{A}$ we have at least three edges $e_1, e_2, e_3 \in E((F_3(P_n))^2)$ such that $e_1 := vx, e_2 := vy, e_3 := xy \in E^*((F_3(P_n))^c)$ with $x, y \in \mathcal{A}$. Then, for every 3 vertices in $\mathcal{A}$ only one can be in an independent set of $(F_3(P_n))^2$. Hence, $\alpha((F_3(P_n))^2) \leq \lceil \alpha(F_3(P_n))/3 \rceil$, as desired.

Since Proposition 5 does not hold for $n \in \{13, 14, 15\}$, we devoted the last part of this section to establishing upper bounds for these three instances.

The next observation will be helpful to find an upper bound of $\rho(F_3(P_n))$ for $n \in \{13, 14, 15\}$.

Observation 2 (Observation 9 [19]). *Let $\{V_1, V_2\}$ be a partition of $V(G)$. If G_1 and G_2 are the subgraphs of G induced by V_1 and V_2, respectively, then $\rho(G) \leq \rho(G_1) + \rho(G_2)$.*

Let us define the vertex set $G_x := \{\{x, i_1, i_2\} : 1 \leq x \leq n-2,\ 2 \leq i_1 \leq n-1,\ 3 \leq i_2 \leq n\} \subseteq V(F_3(P_n))$. Now, we partition $V(F_3(P_n))$ into two subsets V_1 and V_2 as follows. $V_1 = \bigcup_{x=1}^{i} G_x$ and $V_2 = V(F_3(P_n)) \setminus (\bigcup_{x=1}^{i} G_x)$, with $1 \leq i \leq n-2$. Let $\mathcal{G}_{i,n}$ and $\mathcal{G}^*_{i,n}$ be two subgraphs of $F_3(P_n)$ induced by V_1 and V_2, respectively.

It is easy to see that $\mathcal{G}^*_{i,n} \simeq F_3(P_{n-i})$ and $\mathcal{G}_{n-2,n} = F_3(P_n)$. And thus we have the next Proposition 6.

Proposition 6. $\rho(F_3(P_n)) \leq \min\limits_{1 \leq i \leq n-2} \{\rho(\mathcal{G}_{i,n}) + \rho(\mathcal{G}^*_{i,n})\}$.

Proof. This proposition is an immediate consequence of Observation 2.

We obtained $\rho(\mathcal{G}_{3,13}) = 30$, $\rho(\mathcal{G}_{3,14}) = 35$ and $\rho(\mathcal{G}_{3,15}) = 41$ by using *Algorithm 3*. From Table 1 and Proposition 6, it follows that $\rho(F_3(P_{13})) \leq \rho(\mathcal{G}_{3,13}) + \rho(F_3(P_{10})) = 30 + 24 = 54$, $\rho(F_3(P_{14})) \leq \rho(\mathcal{G}_{3,14}) + \rho(F_3(P_{11})) = 35 + 32 = 67$ and $\rho(F_3(P_{15})) \leq \rho(\mathcal{G}_{3,15}) + \rho(F_3(P_{12})) = 41 + 41 = 82$.

In Table 4, we present our lower and upper bounds for $\rho(F_3(P_n))$ with $13 \leq n \leq 32$. The interested reader can find in Table 4 [9] the lower and upper bounds for $\rho(F_3(P_n))$ with $13 \leq n \leq 44$.

Table 4. Our lower and upper bounds for $\rho(F_3(P_n))$ with $13 \leq n \leq 32$.

n	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
Our results of lower bound	50	63	75	91	109	127	149	173	198	226	258	291	327	367	413	455	504	563	612	679
Our results of upper bound	54	67	82	136	163	190	224	257	297	338	386	434	490	546	612	677	752	827	912	998

The interested reader can download from [5] the programs in C++, Python, and Wolfram Mathematica that we have used to obtain these results.

7. Conclusions and future work

In this work, we showed that $\Gamma_n^3 \simeq F_3(P_n)$, and so determining $\rho(F_3(P_n))$ is equivalent to determining the maximum size of a binary code of constant weight 3 which can correct a single adjacent transposition. We have specifically determined the exact value of $\rho(F_3(P_n))$ for $n \leq 12$. However, due to the complexity of the involved calculation of $\rho(F_3(P_n))$ for $n > 12$, we have obtained some lower and upper bounds. On the other hand, since this paper is a first attempt to determine $\rho(F_3(P_n))$ and the difference between the lower and upper bounds of $\rho(F_3(P_n))$ is too small for $n \in \{13, 14, 15\}$, we believe that improving our technique in Section 6 could help to find a tight upper bound for $\rho(F_3(P_n))$. Finally, we also believe that it would be interesting to determine the exact value of $\rho(F_3(P_n))$ for $n > 12$.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

We would like to thank the anonymous referees for careful reading and helpful suggestions.
This research was supported by the Instituto Politécnico Nacional (IPN) through grant SIP/20231701.

Conflict of interest

The authors declare there are no conflicts of interest.

References

1. A. Alzaga, I. Rodrigo, R. Pignol, Spectra of symmetric powers of graphs and the Weisfeiler-Lehman refinements, *J. Comb. Theory B*, **100** (2010), 671–682. https://doi.org/10.1016/j.jctb.2010.07.001
2. S. Butenko, P. Pardalos, I. Sergienko, V. Shylo, P. Stetsyuk, Estimating the size of correcting codes using extremal graph problems, In: *Optimization*, New York: Springer, 2009, 227–243. https://doi.org/10.1007/978-0-387-98096-6_12
3. W. Carballosa, R. Fabila-Monroy, J. Leaños, L. M. Rivera, Regularity and planarity of token graphs, *Discuss. Math. Graph T.*, **37** (2017), 573–586. https://doi.org/10.7151/dmgt.1959
4. H. de Alba, W. Carballosa, J. Leaños, L. M. Rivera, Independence and matching numbers of some token graphs, *Australas. J. Comb.*, **76** (2016), 387–403.
5. J. A. Escareño Fernández, C. Ndjatchi, L. M. Ríos-Castro, Algorithms-for-packing-number-of-3-token, 2024. Available from: `https://github.com/TheAlexz/Algorithms-for-packing-number-of-3-token`.
6. R. Fabila-Monroy, D. Flores Peñaloza, C. Huemer, F. Hurtado, J. Urrutia, D. R. Wood, Token graphs, *Graphs and Combinatorics*, **28** (2012), 365–380. https://doi.org/10.1007/s00373-011-1055-9

7. R. Fabila-Monroy, J. Leaños, A. L. Trujillo-Negrete, On the connectivity of token graphs of trees, *Discrete Math. Theor.*, **24** (2022), 1–23. https://doi.org/10.46298/dmtcs.7538

8. M. R. Garey, D. S. Johnson, *Computers and intractability: A guide to the theory of NP-completeness*, New York: W. H. Freeman, 1979.

9. Google Docs, Algorithms for computing the lower and upper bounds for the packing number of 3-token graph of path graph. Available from: `http://tinyurl.com/25bx8dpe`.

10. A. S. Hassan, A generalisation of Johnson graphs with an application to triple factorisations, *Discrete Math.*, **338** (2015), 2026–2036. https://doi.org/10.1016/j.disc.2015.05.001

11. D. S. Hochbaum, D. B. Shmoys, A best possible heuristic for the k-center problem, *Math. Oper. Res.*, **10** (1985), 175–366. https://doi.org/10.1287/moor.10.2.180

12. J. Leaños, C. Ndjatchi, The edge-cdonnectivity of Token Graphs, *Graph. Combinator.*, **37** (2021), 1013–1023. https://doi.org/10.1007/s00373-021-02301-0

13. J. Leaños, A. L. Trujillo-Negrete, The connectivity of token graphs, *Graph. Combinator.*, **34** (2018), 777–790. https://doi.org/10.1007/s00373-018-1913-9

14. K. G. Mirajkar, Y. B. Priyanka, Traversability and covering invariants of token graphs, *International J. Math. Combin.*, **2** (2016), 132–138.

15. L. M. Riós-Castro, Números de dominación y empaquetamiento de ciertas gráficas de fichas, PhD Thesis, Universidad Autónoma de Zacatecas, 2018.

16. G. Rossum, *Python tutorial*, Netherlands: CWI (Centre for Mathematics and Computer Science), 1995. Available from: `https://dl.acm.org/doi/10.5555/869378`

17. N. J. A. Sloane, On single-deletion-correcting codes, 2002, arXiv: math/0207197. https://doi.org/10.48550/arXiv.math/0207197

18. N. J. A. Sloane, A085680-OEIS. Available from: `https://oeis.org/A085680`.

19. J. M. G. Soto, J. Leaños, L. M. Ríos-Castro, L. M. Rivera, The packing number of the double vertex graph of the path graph, *Discrete Appl. Math.*, **247** (2018), 327–340. https://doi.org/10.1016/j.dam.2018.03.085

20. Wolfram Research, Inc., Mathematica, Version 12.0, Champaign, IL, 2019. Available from: `https://www.wolfram.com/mathematica/`

AIMS Press