



UNIVERSIDAD AUTÓNOMA
METROPOLITANA
DIVISIÓN DE CIENCIAS BÁSICAS E INGENIERÍA

SOBRE GRÁFICAS ITERADAS DE CLANES.

Tesis que presenta
Ismael Ariel Robles Martínez

para obtener el grado de
Doctor en Ciencias (Matemáticas).

Asesor: Dr. Miguel Ángel Pizaña López.

Jurado calificador:

Presidente: Dr. Bernardo LLano Pérez.

Secretario: Dr. Miguel Ángel Pizaña López.

Vocal: Dra. Mika Olsen.

Vocal: Dr. Rafael Villarroel Flores.

Vocal: Dr. José David Flores Peñaloza.

Ciudad de México a 20 de octubre de 2023.



Casa abierta al tiempo

UNIVERSIDAD AUTÓNOMA METROPOLITANA

ACTA DE DISERTACIÓN PÚBLICA

No. 00087

Matrícula: 2142800139

Sobre gráficas iteradas de
clanes.

En la Ciudad de México, se presentaron a las 8:30 horas
del día 20 del mes de octubre del año 2023 en la Unidad
Iztapalapa de la Universidad Autónoma Metropolitana, los
suscritos miembros del jurado:

DR. BERNARDO LLANO PEREZ
DR. RAFAEL VILLARROEL FLORES
DRA. MIKA OLSEN
DR. JOSE DAVID FLORES PEÑALOZA
DR MIGUEL ANGEL PIZANA LOPEZ

Bajo la Presidencia del primero y con carácter de
Secretario el último, se reunieron a la presentación de la
Disertación Pública cuya denominación aparece al margen,
para la obtención del grado de:

DOCTOR EN CIENCIAS (MATEMATICAS)

DE: ISMAEL ARIEL ROBLES MARTINEZ

y de acuerdo con el artículo 78 fracción IV del Reglamento
de Estudios Superiores de la Universidad Autónoma
Metropolitana, los miembros del jurado resolvieron:

Aprobar

Acto continuo, el presidente del jurado comunicó al
interesado el resultado de la evaluación y, en caso
aprobatorio, le fue tomada la protesta.

REVISÓ

MTRA. ROSALIA SERRANO DE LA PAZ
DIRECTORA DE SISTEMAS ESCOLARES

DIRECTOR DE LA DIVISIÓN DE CBI

Román Linares Romero
DR. ROMAN LINARES ROMERO

PRESIDENTE

Bernardo Llano Perez
DR. BERNARDO LLANO PEREZ

VOCAL

Rafael Villarroel Flores
DR. RAFAEL VILLARROEL FLORES

VOCAL

Mika Olsen
DRA. MIKA OLSEN

VOCAL

Jose David Flores Peñaloza
DR. JOSE DAVID FLORES PEÑALOZA

SECRETARIO

Miguel Angel Pizana Lopez
DR MIGUEL ANGEL PIZANA LOPEZ

**SOBRE GRÁFICAS
ITERADAS DE
CLANES**

Universidad Autónoma Metropolitana
Ismael Ariel Robles Martínez
Posgrado en Matemáticas

A mi familia.

AGRADECIMIENTOS

A mi madre y a mis hermanos, por ser una parte fundamental de mi vida y de la historia que me define como persona. Gracias por creer en mí durante todo este tiempo.

A Daniela y Elizabeth, por su invaluable amistad y apoyo. Gracias por haber recorrido el camino de la vida conmigo durante todos estos años y por ser un soporte esencial en este periodo tan difícil que comprendió la pandemia por COVID-19. Gracias por ser y estar.

A Mary, Jessy, Vicky, Alma, Carmen, Julio y Alejandro, por su cálida amistad y camaradería durante nuestros estudios de posgrado. Gracias por las alegrías y risas durante el tiempo que tuvimos el gusto de convivir.

A mi asesor Miguel Pizaña, por su guía, consejos y extenso apoyo durante mis estudios de posgrado. Gracias por tu infinita paciencia, sobre todo durante los dos últimos años de pandemia, en los que este trabajo se detuvo por completo.

A la Universidad Autónoma Metropolitana, por haberme cobijado durante mis estudios de posgrado y por haberme dado la oportunidad de conocer a personas tan diversas.

ÍNDICE GENERAL

1	INTRODUCCIÓN	1
1.1	Breve revisión histórica de la teoría de clanes	1
1.2	Notación y terminología	3
1.3	Resumen de resultados	6
2	GRÁFICAS QUE MAXIMIZAN LA SEGUNDA GRÁFICA DE CLANES	9
2.1	Suspensiones de gráficas y la segunda gráfica de clanes	10
2.2	Gráficas que maximizan a la gráfica de Biclanes	15
2.2.1	El caso par	17
2.2.2	El caso impar	19
2.3	Resultados adicionales y conjeturas	29
3	ALGORITMOS GENÉTICOS	35
3.1	Codificación y decodificación en un algoritmo genético	36
3.2	Función de aptitud	38
3.3	Pasos generales de un algoritmo genético	46
3.3.1	Inicialización	47
3.3.2	Reproducción	47
3.3.3	Cruzamiento	52
3.3.4	Mutación	53
3.3.5	Depredación	54
3.4	Algoritmo genético utilizado	56
3.5	Sobre la gráfica $C_n(1, 3, 6, 7, 8)$	58
4	GRÁFICAS BOBINA GENERALIZADAS	65
4.1	Definición	65
4.2	Resultados generales	68
5	CONCLUSIONES	79
	BIBLIOGRAFÍA	81
	ÍNDICE ALFABÉTICO	87

ÍNDICE DE FIGURAS

Figura 1	Codificación de gráficas en un algoritmo genético.	37
Figura 2	Decodificación en un algoritmo genético.	38
Figura 3	Diagrama de flujo de un algoritmo genético.	47
Figura 4	Segmento para elección aleatoria en un algoritmo genético	49
Figura 5	Gráfica de $\text{Ln}(K^n(G))$.	59
Figura 6	Ejemplo de gráfica bobina con 2 órbitas.	66
Figura 7	Ejemplo de gráfica bobina con 4 órbitas.	67
Figura 8	Clanes representantes de G y $K(G)$, para $G = K(C_n(1, 3, 6, 7, 8))$.	78

ÍNDICE DE TABLAS

Tabla 1	Gráficas G que maximizan $ K^2(G) $, para $2 \leq n \leq 10$.	11
Tabla 2	Gráficas con su función de crecimiento y su aptitud.	44
Tabla 3	Población de ejemplo de un algoritmo genético.	48
Tabla 4	Tabla de valores de $\text{Ln}(K^n(G))$, para $G = C_{25}(1, 3, 6, 7, 8)$.	58

INTRODUCCIÓN

Este documento se encuentra dividido de la siguiente forma:

- En el **capítulo 1**, se presentan la terminología y definiciones básicas que se usan en el resto de este documento. También se presenta un resumen de los resultados derivados de esta investigación.
- En el **capítulo 2**, se muestran los resultados y conjeturas derivados de estudiar el problema de caracterizar las gráficas G , que maximizan el número de vértices de la segunda gráfica iterada de clanes $|K^2(G)|$.
- En el **capítulo 3**, se hace una descripción a extenso detalle de la heurística basada en algoritmos genéticos, para estudiar el problema de determinar si existe una gráfica G , tal que $|K^n(G)| = \Theta(a^n)$, con $a > 1$ y $n > 1$. También se muestran los datos experimentales que nos llevaron conjeturar que $|K^n(C_m(1, 3, 6, 7, 8))| = \Theta(3^{\frac{n}{2}})$, con $m \geq 25$.
- En el **capítulo 4**, se presentan las gráficas bobina generalizadas, que son una generalización de las gráficas iteradas de clanes de la gráfica circulante $C_n(1, 3, 6, 7, 8)$. También se presentan resultados y conjeturas sobre el comportamiento de $|K^n(C_m(1, 3, 6, 7, 8))|$.
- En el **capítulo 5**, se muestran las conclusiones de este trabajo.

1.1 BREVE REVISIÓN HISTÓRICA DE LA TEORÍA DE CLANES

Un *clan* de una gráfica G , es una subgráfica completa maximal. Uno de los primeros trabajos en los que se estudian clanes fue realizado en 1949, por Luce y Perry [43], quienes usaron clanes para estudiar matrices que describen redes sociales. Posteriormente en 1960, Miller y Muller caracterizaron las gráficas que tienen el máximo número de clanes [47]. Esta caracterización también fue demostrada de manera independiente, por Moon y Moser [48], en 1965.

clan de una gráfica

gráficas de
Miller-Muller

En la literatura, las gráficas de n vértices que tienen el mayor número de clanes se les conoce como *gráficas de Miller-Muller*. En [48], Moon y Moser también reportaron cotas para el máximo número de diferentes tamaños de clanes, que pueden ocurrir en una gráfica de n vértices. Un año después, Erdős mejoró las cotas encontradas por Moon y Moser [15].

problema
NP-Completo

Desde entonces, el estudio de los clanes ha estado presente en diversos tópicos de matemáticas discretas. En particular, los clanes han sido de importancia fundamental desde los inicios de la teoría de la complejidad computacional. Por ejemplo, el problema de decisión de determinar si una gráfica tiene un clan de un tamaño dado, es uno de los 21 problemas *NP-Completo*, que se enumeran en el trabajo clásico de Karp [38].

gráfica intersección

La gráfica de clanes $K(G)$ de una gráfica G , es la *gráfica intersección* de los clanes de G (i.e. tiene por vértices a los clanes de G y dos vértices son adyacentes si son distintos y tienen intersección no vacía). Hamelink, en 1968, fue el primero en mostrar una caracterización parcial de las gráficas de clanes [31] (suficiente pero no necesaria). Sin embargo, fueron Roberts y Spencer, quienes en 1971, dieron una de las caracterizaciones más utilizadas a la fecha [57]. Por más de 30 años, la única caracterización que se tenía para gráficas de clanes fue la de Roberts y Spencer. Una caracterización alternativa fue encontrada por Alcón y Gutierrez [1], en el 2003.

operador de clanes
gráficas iteradas de
clanes

Dada una gráfica G , su gráfica de clanes $K(G)$ se puede ver como el resultado de aplicar un operador K , que transforma gráficas en gráficas. A este operador se le conoce como el *operador de clanes*. Al aplicar el operador de clanes de manera iterativa se obtienen las *gráficas iteradas de clanes* de G , las cuales se definen de manera recursiva: esto es, $K^0(G) = G$ y $K^{n+1}(G) = K(K^n(G))$. El estudio de las gráficas iteradas de clanes comenzó en 1972, con el trabajo de Hedetniemi y Slater [34]. A partir de este trabajo seminal, se ha hecho un amplio estudio en las gráficas iteradas de clanes: en particular, Prisner en [54], ha hecho una recopilación de propiedades que comparte el operador de clanes, con otros operadores que resultan de gráficas intersección. También se han encontrado aplicaciones de las gráficas iteradas de clanes en el estudio de la propiedad del punto fijo para conjuntos parcialmente ordenados [33]. En física, se han aplicado en gravedad cuántica de bucles [55, 56].

propiedad de Helly
gráfica clan-Helly

Una propiedad que ha resultado muy útil para estudiar gráficas de clanes, es la *propiedad de Helly* [5]. Se dice que una gráfica G es *clan-Helly*, si para cualquier conjunto $\mathcal{F}$, de clanes de G que se intersectan por pares, se cumple que la intersección de $\mathcal{F}$ es no vacía. En particular, esta propiedad la usó Szwarcfiter para formular uno de los resultados clásicos en la teoría de clanes: el *teorema del triángulo extendido* [61].

teorema del
triángulo extendido

Si $|K^n(G)|$ denota el número de vértices de la n -ésima gráfica de clanes, se dice que G es *clan divergente*, si $\lim_{n \rightarrow \infty} |K^n(G)| = \infty$, en caso contrario se dice que G es *clan convergente*. Decidir el *clan comportamiento* de una gráfica consiste en determinar si la gráfica es clan convergente o clan divergente. En [60], se resumen algunos resultados clásicos sobre el clan comportamiento, como son: la gráfica del octaedro es clan divergente [49], el teorema de la retracción [50] y la caracterización del clan comportamiento para gráficas clan-Helly [2].

clan divergente
clan convergente
clan comportamiento

No se sabe si el clan comportamiento es un *problema decidable*, esto es, que exista un algoritmo que permita determinar si una gráfica es clan convergente o clan divergente. Sin embargo, Cedillo y Pizaña han demostrado que la clan divergencia no es expresable en lógica de primer orden [9] y además no es decidable para el caso de gráficas automáticas (no necesariamente finitas) [10]. Puesto que la decidibilidad del clan comportamiento, no es un problema trivial, en [18-20, 41], se revisan técnicas novedosas para el estudio del clan comportamiento.

problema decidable

Con este trabajo se busca aportar a la teoría de clanes, con el estudio de los siguientes tópicos: gráficas G que maximizan a $|K^2(G)|$; heurísticas basadas en algoritmos genéticos, para buscar gráficas cuya función $f(n) = |K^n(G)|$, sea de interés y finalmente, el estudio de las propiedades bajo el operador de clanes, de una familia de gráficas que hemos denominado gráficas bobina generalizadas.

1.2 NOTACIÓN Y TERMINOLOGÍA

Este documento usa notación estándar de teoría de las gráficas, teoría de la complejidad computacional, álgebra y programación. Las gráficas que se estudian en este trabajo, son gráficas simples y no vacías. También identificamos conjuntos de vértices con sus gráficas inducidas, sin embargo, cuando el contexto lo requiere, se usa $V(G)$, para denotar al conjunto de vértices de la gráfica G .

Si G es una gráfica, el *orden de* G (número de vértices), se denota por $|G|$. El *isomorfismo de gráficas* se denota por $G \cong H$. Si dos vértices x y y , son adyacentes o iguales, se usa la notación $x \simeq y$; de manera similar, se usa $x \sim y$, si los vértices son diferentes y adyacentes. La *vecindad abierta* $N_G(x)$, de un vértice $x \in G$, es el conjunto de todos los vértices de la gráfica G (diferentes de x), que son adyacentes a x . Por otra parte, la *vecindad cerrada* $N_G[x]$, es el conjunto de vértices dado por: $N_G[x] = N_G(x) \cup \{x\}$. Cuando en el contexto sea claro quién es la gráfica G , simplemente escribiremos

orden de una gráfica
isomorfismo de gráficas
vecindad abierta
 $N_G(x)$
vecindad cerrada
 $N_G[x]$

conjunto potencia	$N(x)$ y $N[x]$. El conjunto potencia de los vértices de G (es decir, todos los subconjuntos de $V(G)$), se denota por 2^G .
gráficas K_n, I_n, P_n	Para $n \geq 1$, denotaremos a la <i>gráfica completa</i> , la <i>gráfica sin aristas</i> , la <i>trayectoria</i> y el <i>ciclo</i> de n vértices, por: K_n, I_n, P_n y C_n respectivamente. La gráfica
gráfica $\overline{G}$	<i>complemento</i> de G , se denota por $\overline{G}$. La gráfica que resulta de <i>remover el vértice</i> $x \in G$, de G , se denota por $G - x$. Aunque las gráficas que se estudian
gráfica $G - x$	en este trabajo tienen al menos 1 vértice, resulta conveniente considerar a la <i>gráfica sin vértices</i> , que denotaremos por K_0 , a fin de simplificar algunas
gráfica sin vértices K_0	de las fórmulas que se presentan en el capítulo 2.
suma de Zykov $G + H$	La <i>suma de Zykov</i> $G + H$, de las gráficas G y H , es la gráfica que resulta de la unión disjunta de G y H , agregando todas las posibles aristas entre ellas. En particular, note que $G + K_0 = G$. Si d es un número natural con $d \geq 1$,
octaedro d -dimensional O_d	el <i>octaedro d-dimensional</i> se define como: $O_d = I_2 + \dots + I_2$ (d veces). Por simplicidad, también se usa la notación $d \cdot G = G + \dots + G$ (d veces). De esta forma, el octaedro d -dimensional se puede escribir como: $O_d = d \cdot I_2$. En particular, cuando $d = 0$, se tiene que $O_d = K_0$.
notación O grande	La notación del octaedro no debe de confundirse con la <i>notación O grande</i> , que también se usa en este trabajo para denotar la <i>cota superior asintótica</i> . La
notación Ω	<i>cota inferior asintótica</i> , se denota por Ω . La <i>cota ajustada asintótica</i> , se denota
notación Θ	por Θ , esta se define como $g(n) = \Theta(f(n))$ si y sólo si $g(n) = \Omega(f(n))$ y $g(n) = O(f(n))$.
producto tensorial de gráficas $G \times H$	El <i>producto tensorial</i> $G \times H$ de las gráficas G y H , es la gráfica cuyo conjunto de vértices es el producto cartesiano $V(G) \times V(H)$ y dos vértices (g_1, h_1) y (g_2, h_2) son adyacentes si y sólo si $g_1 \sim g_2$ y $h_1 \sim h_2$.
producto categórico fuerte $G \boxtimes H$	El <i>producto categórico fuerte</i> $G \boxtimes H$ de las gráficas G y H , es la gráfica cuyo conjunto de vértices es el producto cartesiano $V(G) \times V(H)$ y dos vértices (g_1, h_1) y (g_2, h_2) son adyacentes, si y sólo si se cumple alguna de las siguientes condiciones: ▪ $g_1 = g_2$ y $h_1 \sim h_2$, o ▪ $g_1 \sim g_2$ y $h_1 = h_2$.
unión de gráficas $G \cup H$	La <i>unión</i> $G \cup H$ de gráficas G y H , con $V(G) \cap V(H) \neq \emptyset$, es la gráfica con vértices $V(G) \cup V(H)$ y dos vértices son adyacentes, si y sólo si son adyacentes en G o en H . Observe que $\overline{G \cup H} = \overline{G} + \overline{H}$.
grupo Z_n	El <i>grupo de enteros módulo n</i> , se denota por Z_n . Si $[k]$ es una clase lateral de Z_n , se usará la convención de que el representante k , cumple que $0 \leq k \leq$
gráfica circulante	$n - 1$. La <i>gráfica circulante</i> $C_n(a_1, a_2, \dots, a_r)$, con $a_i \in Z_n$, para $1 \leq i \leq r$,

es la gráfica con vértices $V(G) = Z_n$ y dos vértices x y y son adyacentes si y sólo si $x - y \in J = \{\pm a_1, \pm a_2, \dots, \pm a_r\}$.

Como parte de este trabajo, se hicieron múltiples experimentos computacionales para enunciar o refutar conjeturas que surgieron en el desarrollo de la investigación. El código de estos experimentos se implementó usando el *sistema de álgebra computacional* [7]: Groups, Algorithms and Programming (GAP) [23]. También se usó la biblioteca de código de GAP: Yet Another Graph System (YAGS) [8], para simplificar el código necesario para trabajar con gráficas.

sistema GAP
biblioteca YAGS

Los pseudocódigos que se describen en el capítulo 3, hacen uso extenso de la estructura de datos conocida como *lista enlazada*, que es una forma de representar vectores de n entradas, que pueden aumentar o reducir su tamaño dinámicamente. De tal forma que si X es una lista que representa el vector $(a_1, a_2, \dots, a_n)$, entonces $X[1] = a_1$, $X[2] = a_2$ y así sucesivamente. En el contexto de programación, se prefiere la notación $X = [a_1, a_2, \dots, a_n]$, para indicar los valores de X de manera explícita. La notación $|X|$, representa el total de elementos que contiene la lista X .

lista enlazada

En el capítulo 3, también se hace uso del concepto de *función*, en el sentido de *programación estructurada*: es decir, una serie de instrucciones que se ejecutan secuencialmente al "invocar" a la función.

función
(programación)

1.3 RESUMEN DE RESULTADOS

Uno de los problemas que se investigó como parte de este trabajo, fue el caracterizar las gráficas G que maximizan a $|K^2(G)|$. Los resultados obtenidos sobre este problema, se describen a detalle en el capítulo 2 y se publicaron en [52] y en [53]. Si bien, el problema continúa abierto, los experimentos computacionales nos han llevado a conjeturar que si G es una gráfica de n vértices, entonces $|K^2(G)|$ está acotado por los siguientes valores (conjetura 2.36):

$$|K^2(G)| \leq \begin{cases} \sqrt{2}^{\sqrt{2}^n}, & \text{si } n \text{ es par,} \\ \sqrt{5}^{\sqrt{2}^{n-3}}, & \text{si } n \text{ es impar.} \end{cases}$$

También conjeturamos que las cotas superiores se alcanzan con la gráfica $\mathcal{G}_n$, dada por (definición 2.3):

$$\mathcal{G}_n = \begin{cases} O_{\frac{n}{2}}, & \text{si } n \text{ es par,} \\ O_{\frac{n-3}{2}} + I_3, & \text{si } n \text{ es impar.} \end{cases}$$

Si la conjetura fuera cierta, entonces $\mathcal{G}_n$ es una suspensión $S(H)$, de una gráfica H , esto es: $\mathcal{G}_n = S(H) = I_2 + H$. Es por ello que para estudiar las gráficas que maximizan a $|K^2(G)|$, se propuso un nuevo *operador de biclanes* $B(G)$ (definición 2.5), con el cual se caracterizó la segunda gráfica iterada de clanes de suspensiones de G , en términos de $B(G)$, esto es: $B(G) = K^2(S(G))$ (teorema 2.8).

También se caracterizaron las gráficas G que maximizan a $|B(G)|$ y se demostró que esto ocurre precisamente cuando $G = \mathcal{G}_n$, lo que sugiere que la conjetura 2.36, podría ser cierta. En la sección 2.3, presentamos resultados y una estrategia general, que podrían ser útiles para demostrar la conjetura 2.36.

Otro de los problemas que se investigó en este trabajo, fue el determinar si existe una gráfica G , para la que el orden de sus gráficas iteradas de clanes tiene crecimiento exponencial, esto es: $|K^n(G)| = \Theta(a^n)$, con $a > 1$. Este problema lleva a su vez a la cuestión de encontrar una gráfica G , que al menos muestre dicho comportamiento en experimentos computacionales y de esta forma pueda servir como punto de partida de la investigación.

Por las razones que se exponen a detalle en el capítulo 3, no es computacionalmente factible revisar cada gráfica del conjunto de gráficas de n vértices para buscar a la gráfica G de interés, es por ello que se requiere una búsque-

da más dirigida. Para los fines de esta investigación, una heurística basada en algoritmos genéticos arrojó resultados satisfactorios.

Los principios generales de un algoritmo genético, así como la implementación que se usó en este trabajo, se describen con extenso detalle en el capítulo 3. Mediante dicha heurística se encontró que la gráfica circulante $G = C_n(1, 3, 6, 7, 8)$, con $n \geq 25$, *experimentalmente* tiene crecimiento exponencial.

Para estudiar las propiedades de las gráficas $K^i(C_n(1, 3, 6, 7, 8))$, en el capítulo 4, se define una familia de gráficas que hemos denominado *gráficas bobina generalizadas*. Las cuales son una generalización de las gráficas bobina definidas por Larrión, Pizaña y Villarroel-Flores, en [42]. En particular, cualquier gráfica circulante que cumple las condiciones del lema 4.3 (como lo es la gráfica $C_n(1, 3, 6, 7, 8)$), es una gráfica bobina generalizada.

Si G es una gráfica bobina generalizada, en la sección 4.2, se demuestra que $K(G)$ también es una gráfica bobina generalizada (teorema 4.7). Entre otros resultados, también se dan caracterizaciones de las adyacencias de G , en términos de sus clanes (lema 4.12) y se dan lemas para obtener subgráficas completas de $K(G)$.

Tenemos la conjetura de que $G = C_n(1, 3, 6, 7, 8)$, con $n \geq 25$, cumple que $|K^i(G)| = 3 \cdot |K^{i-2}(G)| + |G|$, para $i \geq 2$ (conjetura 3.12). De ser cierta dicha conjetura, en la sección 3.5 se muestra que esto implicaría que $|K^i(G)| = \Theta(3^{\frac{n}{2}})$ (teorema 3.15).

GRÁFICAS QUE MAXIMIZAN LA SEGUNDA GRÁFICA DE CLANES

Si G es una gráfica de n vértices, una pregunta natural que surge al estudiar el operador de clanes es la siguiente: ¿qué gráficas G maximizan $|K^i(G)|$, para $i \geq 1$?

Para $i = 1$, es bien conocida la caracterización de las gráficas G que maximizan a $|K(G)|$. Dicha caracterización fue descubierta de manera independiente, primero por Miller y Muller en 1960 [47] y posteriormente por Moon y Moser en 1965 [48]. En dichos trabajos, se demuestra el siguiente teorema, el cual da una cota superior para el número máximo de clanes que puede tener una gráfica de n vértices (en el 2011, una prueba más simple fue publicada por Wood [65]).

Teorema 2.1 (Miller y Muller [47, Teorema 4], Moon y Moser [48, Teorema 1], Wood [47]). *Sea G una gráfica con $n \geq 2$ vértices, entonces:*

$$|K(G)| \leq g(n) = \begin{cases} 3^{n/3}, & \text{si } n \equiv 0 \pmod{3} \\ 4 \cdot 3^{(n-4)/3}, & \text{si } n \equiv 1 \pmod{3} \\ 2 \cdot 3^{(n-2)/3}, & \text{si } n \equiv 2 \pmod{3}. \end{cases}$$

Las gráficas que realizan las cotas del Teorema 2.1, se les conoce como *gráficas de Miller-Muller*. A fin de simplificar la descripción de dichas gráficas, recuerde que para un entero $d \geq 0$ y una gráfica G , usaremos la notación $d \cdot G = G + G + \dots + G$ (d veces G). Las gráficas de Miller-Muller se describen en el siguiente teorema.

*gráficas de
Miller-Muller*

Teorema 2.2 (Miller y Muller [47, Teoremas 11, 12 y 13], Moon y Moser [48, Teorema 2]). *Sea $n \geq 2$, sea $g(n)$ la función definida en el Teorema 2.1 y sea M_n las gráficas de n vértices que se definen como:*

$$M_n = \begin{cases} \frac{n}{3} \cdot I_3, & \text{si } n \equiv 0 \pmod{3} \\ H + \frac{n-4}{3} \cdot I_3, \text{ donde } H \in \{C_4, I_4\}, & \text{si } n \equiv 1 \pmod{3} \\ I_2 + \frac{n-2}{3} \cdot I_3, & \text{si } n \equiv 2 \pmod{3}. \end{cases}$$

Entonces $|K(G)| = g(n)$, sólo cuando $G = M_n$.

Para $i \geq 2$, **sigue siendo un problema abierto qué gráficas maximizan $|K^i(G)|$** . A fin de avanzar en la solución de dicho problema, en este trabajo de investigación doctoral definimos un operador de biclanes B , que para la suspensión $S(G) := G + I_2$, de la gráfica G , tiene la propiedad de que $K^2(S(G)) \cong B(K(G))$. También caracterizamos las gráficas que maximizan $|B(G)|$. Estos resultados se presentaron en [52] y en [53], mismos que se describen a detalle en las siguientes secciones.

2.1 SUSENSIONES DE GRÁFICAS Y LA SEGUNDA GRÁFICA DE CLANES

Puesto que las gráficas de Miller-Muller maximizan el orden de la primera gráfica de clanes, se podría esperar que dichas gráficas también maximizan el orden de la segunda gráfica de clanes. Sin embargo, para $n \geq 2$ vértices, nuestros experimentos computacionales muestran que este no es el caso. Para un entero $d \geq 0$ y considerando $O_d = d \cdot I_2 = I_2 + I_2 + \cdots + I_2$ (d veces). Los experimentos muestran que las siguientes gráficas tienen mayor número de clanes que las gráficas de Miller-Muller (véase Tabla 1).

Gráfica $\mathcal{G}_n$ **Definición 2.3.** Sea $n \geq 2$ vértices, definimos a las gráficas $\mathcal{G}_n$, como:

$$\mathcal{G}_n = \begin{cases} O_{\frac{n}{2}}, & \text{si } n \text{ es par,} \\ O_{\frac{n-3}{2}} + I_3, & \text{si } n \text{ es impar.} \end{cases}$$

Para $2 \leq n \leq 5$, los resultados experimentales que se reportan en la Tabla 1, muestran que $|K^2(G)|$ se maximiza cuando G es alguna de las gráficas: $\mathcal{G}_4$ o I_4 . Para $n = 5$, las gráficas que maximizan son: $\mathcal{G}_5$, I_5 , C_5 y $C_4 \cup I_1$. Sin embargo, para $n \geq 6$, los experimentos indican que las gráficas $\mathcal{G}_n$ son las únicas que maximizan $|K^2(G)|$.

En la Tabla 1, también se comparan el número de clanes de clanes que tienen las gráficas $\mathcal{G}_n$ y las gráficas de Miller-Muller M_n (columnas $|K^2(\mathcal{G}_n)|$ y $|K^2(M_n)|$, respectivamente). Note que para $n = 6$, el valor de $|K^2(\mathcal{G}_n)|$, supera por más del triple al valor de $|K^2(M_n)|$. Esta diferencia se hace aún mayor a medida que n aumenta.

Para nuestros experimentos usamos GAP [23], YAGS [8] y la lista de gráficas simples de Brendan McKay [46]. Es interesante notar que estos experimentos sólo pueden realizarse para gráficas G , con un número de vértices n , en un rango limitado de valores (alrededor de $n \leq 18$). En parte, esto se debe a que la segunda gráfica de clanes de un octaedro d -dimensional, cumple que [49]:

$$|K^2(O_d)| = 2^{2^{d-1}}.$$

Tabla 1: Gráficas G que maximizan $|K^2(G)|$, para $2 \leq n \leq 10$.

n	G	$ K^2(G) $	$ K^2(M_n) $
2	$\mathcal{G}_2$	2	2
3	$\mathcal{G}_3$	3	3
4	$\mathcal{G}_4$ y I_4	4	4
5	$\mathcal{G}_5, C_4 \cup I_1, C_5$ y I_5	5	5
6	$\mathcal{G}_6$	16	6
7	$\mathcal{G}_7$	25	7
8	$\mathcal{G}_8$	256	44
9	$\mathcal{G}_9$	625	90
10	$\mathcal{G}_{10}$	65536	154

Por lo que la gráfica G que maximiza $|K^2(G)|$, debe cumplir que:

$$|K^2(G)| \geq |K^2(O_d)| \geq 2^{2^{d-1}}, \text{ con } d = \left\lfloor \frac{n}{2} \right\rfloor,$$

lo que nos da un orden super exponencial. Por ejemplo, para $n = 18$, $K(G)$ *debe* tener un número de clanes mayor o igual que $2^{2^8} = 2^{256}$, el cual es un número mucho más grande que 2^{90} , el cual es el total de átomos que se estima que tiene el universo visible [30]. Sin un conocimiento de la estructura de G , es imposible almacenar esa cantidad de información para representar $K^2(G)$. Más aún, sin considerar las limitaciones de memoria y si se usara una de las supercomputadoras más rápidas al momento (que pueden realizar $2 \cdot 10^{17}$ operaciones por segundo [63]) y suponiendo que se pudiera calcular 1 clan, en 1 operación (una estimación irreal y ridículamente optimista), llevaría más de 53 billones de años calcular los clanes de clanes de G .

A pesar de que los experimentos que se pueden realizar son limitados, estos nos permitieron conjeturar que $|K^2(G)|$ se maximiza cuando $G = \mathcal{G}_n$. Los experimentos también sugieren que $|K^2(G)|$ es mayor, cuando G es una *suspensión*, que cuando no lo es. En este trabajo diremos que G es una suspensión de una gráfica H , si $G \cong S(H) := H + I_2$. De esto se tiene que para $n \geq 4$, la gráfica $\mathcal{G}_n$ es una suspensión. Por lo que si nuestra conjetura es cierta, es de esperarse que $|K^2(S(G))|$ se maximiza cuando $S(G) = \mathcal{G}_{|G|+2}$. A fin de estudiar $|K^2(S(G))|$, haremos uso de las siguientes definiciones.

suspensión de una gráfica

Definición 2.4. Biclan de una gráfica. Sea G una gráfica simple y sea

$$\mathcal{B} = \{(X, Y) \in 2^G \times 2^G : x \simeq y, \text{ para cada } x \in X \text{ y } y \in Y\}.$$

Defina un orden parcial en $\mathcal{B}$ con $(X_1, Y_1) \preceq (X_2, Y_2)$ si y sólo si $X_1 \subseteq X_2$ y $Y_1 \subseteq Y_2$. Un biclan (X, Y) de G , es un elemento maximal de $\mathcal{B}$ bajo el orden $\preceq$.

Definición 2.5. Gráfica de biclanes. La gráfica de biclanes $B(G)$ de una gráfica G , es la gráfica cuyos vértices son biclanes de G y dos vértices diferentes (X_1, Y_1) y (X_2, Y_2) de $B(G)$, son adyacentes si y sólo si $X_1 \cap X_2 \neq \emptyset$ o $Y_1 \cap Y_2 \neq \emptyset$.

La Definición 2.4 nos dice que un biclan (X, Y) , es un par ordenado de subconjuntos de vértices no necesariamente disjuntos de G , tales que cada vértice en X es adyacente o igual a cada vértice en Y y (X, Y) es maximal bajo inclusión de componentes.

Múltiples variaciones del operador de biclanes han sido estudiadas por Prisner [54], Figueroa y Llano [16], Zelinka [66] y Groshaus et al [25-29]. Las definiciones 2.4 y 2.5, hasta donde sabemos son nuevas en la literatura, publicadas en [52] y [53].

Observe que si $S(G)$ es la suspensión de la gráfica G , por definición existen vértices diferentes $x, y \in S(G) \setminus G$, tales que $N_{S(G)}(x) = N_{S(G)}(y) = G$ (son adyacentes a todos los vértices de G). Esta observación y los siguientes resultados, permiten relacionar a $K^2(S(G))$ con $B(K(G))$.

Lema 2.6. Sean q y p clanes de G y $S(G)$, respectivamente. Sean los vértices $x, y \in S(G) \setminus G$, entonces se cumplen las siguientes afirmaciones:

- (i). $p \setminus \{x, y\} \in K(G)$.
- (ii). Para $q_x = q \cup \{x\}$ y $q_y = q \cup \{y\}$, se tiene que $q_x, q_y \in K(S(G))$.

Demostración. Para (i), como $N_{S(G)}(x) = N_{S(G)}(y) = G$, si p es un clan de $S(G)$, entonces $x \in p$ o $y \in p$ y por la maximalidad de p , se tiene que $p \setminus \{x, y\}$ es un clan de G .

Para (ii), si $q \in K(G)$, como $N_{S(G)}(x) = N_{S(G)}(y) = G$, entonces $q_x = q \cup \{x\}$ y $q_y = q \cup \{y\}$, son completas de $S(G)$ y por la maximalidad de q , se tiene que $q_x, q_y \in K(S(G))$. $\square$

Lema 2.7. Sean los vértices $x, y \in S(G) \setminus G$, entonces $(X, Y) \in B(K(G))$ si y sólo si la colección:

$$Q = \left(\bigcup_{q \in X} \{q \cup \{x\}\} \right) \cup \left(\bigcup_{q' \in Y} \{q' \cup \{y\}\} \right),$$

es un clan de $K(S(G))$.

Demostración. Sea (X, Y) un biclan de $K(G)$, por definición de biclan, para cualquier $q \in X$ y $q' \in Y$, se tiene que $q \simeq q'$ en $K(G)$, luego $q \cap q' \neq \emptyset$. Además, por el Lema 2.6, inciso (ii), para $q_x = q \cup \{x\}$ y $q'_y = q' \cup \{y\}$, se tiene que $q_x, q'_y \in K(S(G))$, luego:

$$Q = \left(\bigcup_{q \in X} \{q_x\} \right) \cup \left(\bigcup_{q' \in Y} \{q'_y\} \right),$$

es una completa de $K(S(G))$. Para mostrar que Q también es clan de $K(S(G))$, sea $p \in K(S(G))$, tal que $Q \cup \{p\}$ es una completa de $K(S(G))$. Sin pérdida de generalidad, se tiene que $x \in p$ y por el Lema 2.6, inciso (i), se cumple que $q'' = p \setminus \{x\} \in K(G)$. Como Q es una completa, para cualquier $q' \in Y$, se cumple que $p \simeq q'_y = q' \cup \{y\}$ en $K(S(G))$. De esto se tiene que $q'' \simeq q'$ en $K(G)$ y por la maximalidad del biclan (X, Y) , se cumple que $q'' \in X$. Por lo tanto $Q \cup \{p\} = Q$ y en consecuencia Q es un clan de $K(S(G))$.

Por otra parte, sean los conjuntos $X, Y \subseteq K(G)$ tales que la colección:

$$Q = \left(\bigcup_{q \in X} \{q_x\} \right) \cup \left(\bigcup_{q' \in Y} \{q'_y\} \right),$$

con $q_x = q \cup \{x\}$ y $q'_y = q' \cup \{y\}$, es un clan de $K(S(G))$. Entonces para cualquier $q \in X$ y $q' \in Y$, se tiene que $q_x \sim q'_y$ en $K(S(G))$, luego $q \simeq q'$ en $K(G)$. De esto se tiene que el par ordenado $W = (X, Y)$, es una bicompleta de $K(G)$.

Para mostrar que W es biclan de $K(G)$. Sea $q'' \in K(G)$, tal que $W' = (X \cup \{q''\}, Y)$, es una bicompleta de $K(G)$. Sea $q''_x = q'' \cup \{x\}$, entonces para cualquier $q \in X$, se cumple que $x \in q''_x \cap q_x$, lo cual implica que $q''_x \simeq q_x$ en $K(S(G))$. Además, para cualquier $q' \in Y$, se cumple que $q'' \simeq q'$ en $K(G)$, por lo que $q''_x \sim q'_y$ en $K(S(G))$. De esto se tiene que $Q \cup \{q''\}$ es una completa de $K(S(G))$, pero dado que Q es un clan, entonces $Q = Q \cup \{q''\}$ y en consecuencia $q'' \in X$. De esto se concluye que $W = W'$, por lo que W es biclan de $K(G)$.

□

Teorema 2.8. Sea G una gráfica simple. Entonces $K^2(S(G)) \cong B(K(G))$.

Demostración. Sean los vértices $x, y \in S(G) \setminus G$. Sea $(X, Y) \in B(K(G))$, por el Lema 2.7, es posible definir un mapeo $\tau : B(K(G)) \rightarrow K^2(S(G))$ dado por:

$$\tau((X, Y)) = \left(\bigcup_{q \in X} \{q \cup \{x\}\} \right) \cup \left(\bigcup_{q' \in Y} \{q' \cup \{y\}\} \right).$$

Se mostrará que τ es un isomorfismo de gráficas.

Para mostrar que τ es sobreyectiva, sea Q un clan de $K(S(G))$. Por el Lema 2.6, si $p \in Q$, entonces $p \setminus \{x, y\} \in K(G)$, por lo que existen conjuntos $X, Y \subseteq K(G)$, tales que

$$Q = \left(\bigcup_{q \in X} \{q_x\} \right) \cup \left(\bigcup_{q' \in Y} \{q'_y\} \right),$$

con $q_x = q \cup \{x\}$ y $q'_y = q \cup \{y\}$. Además, por el Lema 2.7, se tiene que $W = (X, Y) \in B(K(G))$. Se sigue que $\tau(W) = Q$ y por lo tanto τ es sobre.

Para el resto de esta prueba $W_1 = (X_1, Y_1)$ y $W_2 = (X_2, Y_2)$, son biclanes de $K(G)$.

Para probar que τ es inyectiva, si $\tau(W_1) = \tau(W_2)$, entonces para $A_1 = \bigcup_{q \in X_1} \{q \cup \{x\}\}$ y $A_2 = \bigcup_{q \in X_2} \{q \cup \{x\}\}$, se tiene que $A_1 = A_2$, luego $X_1 = X_2$. Un argumento similar muestra que $Y_1 = Y_2$. En consecuencia $W_1 = W_2$ y τ es inyectiva.

Finalmente, para mostrar que τ preserva adyacencias, se tienen dos casos:

- Si $W_1 \simeq W_2$ en $B(K(G))$, por definición de gráfica de biclanes, se tiene que $X_1 \cap X_2 \neq \emptyset$ o $Y_1 \cap Y_2 \neq \emptyset$, por lo que existe un clan $q \in K(G)$, tal que $q \in X_1 \cap X_2$ o $q \in Y_1 \cap Y_2$. Además, por el Lema 2.6, inciso (ii), se cumple que $q_x = q \cup \{x\}$ y $q_y = q \cup \{y\}$, son clanes de $S(G)$. Por lo que $q_x \in \tau(W_1) \cap \tau(W_2)$ o $q_y \in \tau(W_1) \cap \tau(W_2)$, lo cual implica que $\tau(W_1) \cap \tau(W_2) \neq \emptyset$ y por lo tanto $\tau(W_1) \simeq \tau(W_2)$ en $K^2(S(G))$.
- Si $\tau(W_1) \simeq \tau(W_2)$ en $K^2(S(G))$, existe un clan $p \in K(S(G))$, tal que $p \in \tau(W_1) \cap \tau(W_2)$. Por el Lema 2.6, inciso (i), se tiene que $q = p \setminus \{x, y\} \in K(G)$, por lo que $q \in X_1 \cap X_2$ o $q \in Y_1 \cap Y_2$ y por la definición de gráfica de biclanes, esto implica que $W_1 \simeq W_2$ en $B(K(G))$.

De esto se concluye que τ es un isomorfismo de gráficas. □

Puesto que el Teorema 2.8, muestra una relación estrecha entre la segunda gráfica de clanes y el operador de biclanes, resulta natural preguntarse qué gráficas G , maximizan $|B(G)|$. La caracterización de dichas gráficas, se estudia a detalle en la siguiente sección.

2.2 GRÁFICAS QUE MAXIMIZAN A LA GRÁFICA DE BICLANES

Si $G \times H$ denota el producto tensorial de gráficas y $\overline{G}$ denota la gráfica complemento de G , el *producto bola* de G con H se define como:

producto bola

$$G \circ H = \overline{\overline{G} \times \overline{H}}.$$

Observe que en $G \circ H$, dos vértices diferentes (g_1, h_1) y (g_2, h_2) son adyacentes, si y sólo si $g_1 \simeq g_2$ en G o $h_1 \simeq h_2$ en H . En la literatura, a veces se usa la notación $\overline{\times}$, para denotar al producto bola [32].

Es bien sabido que la gráfica de clanes de una suma de Zykov es el producto bola de sus gráficas de clanes: $K(G + H) \cong K(G) \circ K(H)$ [40, 49, 50]. El siguiente teorema, muestra que el operador de biclanes también tiene esta propiedad.

Teorema 2.9. *Sean G y H gráficas simples, entonces $B(G + H) \cong B(G) \circ B(H)$.*

Demostración. Sea (X, Y) un biclan de $G + H$, entonces existen subconjuntos de vértices G_1 y G_2 de G y subconjuntos de vértices H_1 y H_2 de H , tales que $X = G_1 \cup H_1$ y $Y = G_2 \cup H_2$. Por la maximalidad de (X, Y) , los pares ordenados (G_1, G_2) y (H_1, H_2) son biclanes de G y H respectivamente. Más aún, por la definición del producto bola, el par ordenado $((G_1, G_2), (H_1, H_2))$ es un vértice de $B(G) \circ B(H)$. Estas observaciones permiten definir la función $\phi : B(G + H) \rightarrow B(G) \circ B(H)$ como sigue:

$$\phi((G_1 \cup H_1, G_2 \cup H_2)) = ((G_1, G_2), (H_1, H_2)).$$

Mostraremos que ϕ es un isomorfismo de gráficas. Sea $T = ((G_1, G_2), (H_1, H_2))$ un vértice de $B(G) \circ B(H)$, entonces $W = (G_1 \cup H_1, G_2 \cup H_2)$ es un biclan de $G + H$ que satisface $\phi(W) = T$. Luego ϕ es sobreyectiva.

Para el resto de esta prueba, sean $(X, Y) = (G_1 \cup H_1, G_2 \cup H_2)$ y $(X', Y') = (G'_1 \cup H'_1, G'_2 \cup H'_2)$, biclanes de $G + H$.

Si $\phi((X, Y)) = \phi((X', Y'))$, de la definición de ϕ , tenemos que $((G_1, G_2), (H_1, H_2)) = ((G'_1, G'_2), (H'_1, H'_2))$. Entonces $G_1 = G'_1$, $G_2 = G'_2$, $H_1 = H'_1$ y $H_2 = H'_2$, por lo tanto $X = X'$ y $Y = Y'$. Luego $(X, Y) = (X', Y')$ y de esto se tiene que ϕ es inyectiva.

Finalmente, para mostrar que ϕ preserva adyacencias, note que $(X, Y) \simeq (X', Y')$ en $B(G + H)$, si y sólo si $(G_1 \cup H_1) \cap (G'_1 \cup H'_1) \neq \emptyset$ o $(G_2 \cup H_2) \cap (G'_2 \cup H'_2) \neq \emptyset$, lo cual es cierto si y sólo si $(G_1, G_2) \simeq (G'_1, G'_2)$ en $B(G)$ o $(H_1, H_2) \simeq (H'_1, H'_2)$ en $B(H)$, esto es $\phi(W_1) \simeq \phi(W_2)$ en la gráfica

$B(G) \circ B(H)$. Por lo tanto, ϕ preserva adyacencias. Concluimos que ϕ es un isomorfismo de gráficas. $\square$

Dada una gráfica G y un subconjunto S de vértices de G , se define la *vecindad cerrada de un conjunto de vértices* como

$$N[S] = \begin{cases} \bigcap_{x \in S} N_G[x], & \text{si } S \neq \emptyset, \\ V(G), & \text{si } S = \emptyset. \end{cases}$$

Es decir, $N[S]$ es el conjunto de todos los vértices que son adyacentes o iguales a todos los vértices de S . Podemos iterar al operador N , definiendo $N^0[S] = S$ y $N^{n+1}[S] = N[N^n[S]]$. Algunas de las propiedades del operador N , se muestran a continuación.

Teorema 2.10. *Para cualquier par de subconjuntos de vértices A y B de G , se cumplen las siguientes propiedades:*

- a) $N[A \cup B] = N[A] \cap N[B]$.
- b) Si $A \supseteq B$, entonces $N[A] \subseteq N[B]$.
- c) $N^2[A] \supseteq A$.
- d) $N^n[A] = N^{n-2}[A]$, para $n \geq 3$.

Demostración. a) y b) son inmediatos de la definición de $N[X]$. Para c), note que cualquier par de vértices $x \in N[A]$ y $y \in A$, satisface que $x \simeq y$, en consecuencia $N^2[A] \supseteq A$.

Para d), la prueba es por inducción. Si $n = 3$, suponga que la afirmación es falsa, entonces existe un vértice $w \in N^3[A] \setminus N[A]$ y por lo tanto, para todo $y \in N^2[A]$ se tiene que $w \simeq y$. Pero por el inciso c), tenemos que $N^2[A] \supseteq A$, luego w es adyacente o igual a todo vértice en A , pero esto implica que $w \in N[A]$, una contradicción. Para el paso inductivo, sea $n \geq 5$, entonces

$$N^{n+1}[X] = N[N^n[X]] = N[N^{n-2}[X]] = N^{n-1}[X] = N^{(n+1)-2}[X].$$

$\square$

La relación que hay entre los biclanes y la vecindad cerrada de un conjunto de vértices, se muestra en el siguiente teorema.

Lema 2.11. *(X, Y) es un biclan de G si y sólo si $X = N[Y]$ y $Y = N[X]$.*

Demostración. Se mostrará que $X = N[Y]$, para ello sea $x \in X$, por definición de biclan, para todo $y \in Y$ se tiene que $x \simeq y$, por lo que $x \in N[Y]$ y $X \subseteq N[Y]$. Para la otra contención, sea $x' \in N[Y]$, por definición de $N[Y]$, para todo $y \in Y$, se cumple que $x' \simeq y$ y por la maximalidad de (X, Y) , se tiene que $x' \in X$, luego $N[Y] \subseteq X$. Se concluye que $X = N[Y]$.

Un argumento similar muestra que $Y = N[X]$. □

Teorema 2.12. (X, Y) es un biclan de G si y sólo si existe un conjunto $S \subseteq G$, tal que $X = N^2[S]$ y $Y = N[S]$.

Demostración. Sea (X, Y) es un biclan de G , por el Lema 2.11, se tiene que $(X, Y) = (N[Y], N[X]) = (N[N[X]], N[X]) = (N^2[X], N[X])$. Luego $S = X$.

Por otra parte, si $S \subseteq G$, sean $Y = N[S]$ y $X = N^2[S]$. Entonces $X = N[N[S]] = N[Y]$ y por el Lema 2.10, inciso d), se tiene que $N[X] = N[N^2[S]] = N^3[S] = N[S] = Y$. Como $X = N[Y]$ y $Y = N[X]$, del Lema 2.11 se concluye que (X, Y) es un biclan de G . □

El teorema 2.12, permite definir la siguiente función:

Definición 2.13. Definimos la función de biclanes $\beta : 2^G \rightarrow B(G)$ como

función de biclanes
 β

$$\beta(S) = (N^2[S], N[S]).$$

La función de biclanes β resulta útil para obtener cotas superiores para el número máximo de biclanes que una gráfica puede tener, tal como se muestra en el siguiente corolario.

Corolario 2.14. La función de biclanes β es sobreyectiva. En particular:

$$|B(G)| = |\{\beta(S) : S \in 2^G\}| \leq 2^{|G|}.$$

Demostración. Inmediato del Teorema 2.12. □

2.2.1 El caso par

A continuación se estudian las gráficas G , que maximizan a $|B(G)|$, cuando $n = |G|$, es par. Por el Corolario 2.14, se tiene que $|B(G)| \leq 2^{|G|}$. El siguiente

teorema muestra que la igualdad se alcanza, justo cuando G es la gráfica del octaedro d -dimensional.

Teorema 2.15. *Las siguientes afirmaciones son equivalentes:*

1. La función β es inyectiva.
2. $N[X] \neq N[X']$ para todo $X, X' \subseteq G$ con $X \neq X'$.
3. $N[G \setminus \{x\}] \neq N[G]$ para todo $x \in G$.
4. Para todo $x \in G$, existe algún $y \in G$ tal que $x \not\succeq y$ y $y \simeq z$ para todo $z \in G \setminus \{x\}$.
5. $n = |G|$ es par y $G \cong O_d$, para $d = \frac{n}{2}$.

Demostración. Las implicaciones (1) $\Rightarrow$ (2) $\Rightarrow$ (3) son inmediatas. Para (3) $\Rightarrow$ (4), como $N[G \setminus \{x\}] \neq N[G]$ y $N[G] \subseteq N[G \setminus \{x\}]$, debe de haber al menos un vértice y , en el conjunto $N[G \setminus \{x\}] \setminus N[G]$. Note que $x \not\succeq y$, de lo contrario se tendría $y \in N[G]$, una contradicción.

Para (4) $\Rightarrow$ (5), note que x es el único vértice de G que satisface $y \not\succeq x$, por lo tanto $x \simeq z$, para todo $z \in G \setminus \{y\}$. De esto se tiene que $G \cong I_2 + I_2 + \cdots + I_2$ (d veces).

Finalmente, para (5) $\Rightarrow$ (1), sean X_1 y X_2 , subconjuntos de vértices diferentes de O_d . Sin pérdida de generalidad, suponga que $X_1 \setminus X_2 \neq \emptyset$, entonces existe un vértice $z \in X_1 \setminus X_2$ y un vértice $w \in O_d$, tal que $w \not\succeq z$. Note que $N[w] = G \setminus \{z\}$, luego $w \in N[X_2]$ y $w \notin N[X_1]$. Por lo tanto $N[X_1] \neq N[X_2]$ y β es inyectiva. $\square$

Del Teorema 2.15, se sigue la siguiente caracterización de las gráficas que maximizan el número de biclanes, cuando el número de vértices de la gráfica es par.

Teorema 2.16. *Sea G una gráfica con orden $n = 2d$ y $d \geq 1$. Si G maximiza $|B(G)|$, entonces $|B(G)| = 2^n$ y $G \cong O_d$.*

Demostración. Del Corolario 2.14, se tiene que $|B(G)| = |\{\beta(S) : S \in 2^G\}| \leq 2^n$.

Si $|B(G)| = 2^n$, entonces $|\{\beta(S) : S \in 2^G\}| = |2^G| = 2^n$, por lo que la función β es inyectiva. Pero de las equivalencias 1 y 5 del Teorema 2.15, esto implica que $G \cong O_d$, con $d = \frac{n}{2}$. $\square$

2.2.2 El caso impar

Se mostrará que cuando $n = |G| \geq 3$ es impar, entonces $|B(G)|$ se maximiza cuando $G \cong I_3 + O_d$, con $d = \frac{n-3}{2}$. Para ello, primero se calculará el valor de $|B(I_3 + O_d)|$, a fin de obtener una cota inferior para $\max\{|B(G)| : |G| = n\}$.

Lema 2.17. *Sea $n = 2d + 3 = |I_3 + O_d|$, entonces $|B(I_3 + O_d)| = \frac{5}{8} \cdot 2^n$.*

Demostración. Note que los biclanes de I_3 , son: $(\emptyset, I_3)$, $(I_3, \emptyset)$ y $(\{x\}, \{x\})$, para cada $x \in I_3$; luego $|B(I_3)| = 5$. Para $n \geq 5$, por el Teorema 2.9, se tiene que $B(I_3 + O_d) \cong B(I_3) \circ B(O_d)$ y por el Teorema 2.16, $|B(O_d)| = 2^{2d}$, luego:

$$|B(I_3 + O_d)| = |B(I_3) \circ B(O_d)| = |B(I_3)| \cdot |B(O_d)| = 5 \cdot 2^{2d} = \frac{5}{8} \cdot 2^n.$$

□

Recuerde que la función β (Definición 2.13), cumple que para cualquier subconjunto de vértices $S \subseteq G$, su imagen $\beta(S)$ es un biclan. Dada una colección $\mathcal{C}$ de subconjuntos de vértices de G , se usará la función β para definir el conjunto de biclanes $\mathbb{B}(\mathcal{C})$ inducido por $\mathcal{C}$, como

conjunto de biclanes
 $\mathbb{B}(\mathcal{C})$ inducido por $\mathcal{C}$

$$\mathbb{B}(\mathcal{C}) = \{\beta(S) : S \in \mathcal{C}\}.$$

Por el Teorema 2.12, los biclanes de una gráfica G se pueden obtener aplicando la función β , a cada subconjunto de vértices de G , es decir, $B(G) = \mathbb{B}(2^G)$.

Note que $|\mathbb{B}(\mathcal{C})| \leq |\mathcal{C}|$. Una cota superior alternativa para $|\mathbb{B}(\mathcal{C})|$, que en ocasiones puede ser más útil, es la que se describe en el siguiente lema.

Lema 2.18. *Sea $S \subseteq G$ y sea $\mathcal{C}$ una colección de subconjuntos de G , tal que para cada $X \in \mathcal{C}$, se tiene que $N[X] \subseteq S$, entonces $|\mathbb{B}(\mathcal{C})| \leq 2^{|S|}$.*

Demostración. Por el Teorema 2.12, para cada $Z \subseteq G$, el biclan $(N^2[Z], N[Z])$ está determinado de manera única por el subconjunto $N[Z] \subseteq G$, por lo que $|\mathbb{B}(\mathcal{C})| = |\{N[X] : X \in \mathcal{C}\}|$. Como para cada $X \in \mathcal{C}$, se tiene que $N[X] \subseteq S$, se sigue que:

$$|\mathbb{B}(\mathcal{C})| = |\{N[X] : X \in \mathcal{C}\}| \leq 2^{|S|}.$$

□

- gráfica* $\mathcal{G}$ Denotaremos por $\mathcal{G}$, a la gráfica que maximiza $|B(G)|$, cuando $|G| = n = 2d + 3$.
- colección de conjuntos* S_x^G Denotaremos por S_x^G , a la colección de subconjuntos de vértices de G , que contienen al vértice x . Análogamente, $S_{\bar{x}}^G$ es la colección de subconjuntos de G , que **no** contienen al vértice x . La colección de subconjuntos que contienen al vértice x y que no contienen al vértice y , se denotará por $S_{x, \bar{y}}^G$ y así sucesivamente. Cuando en el contexto sea claro quien es G , omitiremos el superíndice. Note que $2^G = S_x \cup S_{\bar{x}}$ y $B(G) = \mathbb{B}(2^G) = \mathbb{B}(S_x) \cup \mathbb{B}(S_{\bar{x}})$.
- vértice universal* Se dice que $x \in G$, es un *vértice universal* de G , si $N[x] = G$. El siguiente lema muestra que si G tiene vértices universales, entonces G tiene menos biclanes que $I_3 + O_d$, por lo que $G \neq \mathcal{G}$.

Lema 2.19. *La gráfica $\mathcal{G}$ no tiene vértices universales.*

Demostración. Suponga que $\mathcal{G}$ tiene un vértice universal x . Defina el mapeo $\kappa : S_x \rightarrow S_{\bar{x}}$ como $\kappa(S) = S \setminus \{x\}$. Observe que κ es una biyección, más aún, para cualquier conjunto $X \in S_x$, se tiene que:

$$N[X] = N[X \setminus \{x\}] \cap N[x] = N[X \setminus \{x\}] \cap \mathcal{G} = N[X \setminus \{x\}] = N[\kappa(X)],$$

por lo cual, $\mathbb{B}(S_x) = \mathbb{B}(S_{\bar{x}})$. Además, como $2^{\mathcal{G}} = S_x \cup S_{\bar{x}}$ y $B(\mathcal{G}) = \mathbb{B}(2^{\mathcal{G}})$, el número de biclanes de $\mathcal{G}$ se puede acotar como sigue:

$$\begin{aligned} |B(\mathcal{G})| &= |\mathbb{B}(2^{\mathcal{G}})| = |\mathbb{B}(S_x \cup S_{\bar{x}})| = |\mathbb{B}(S_x) \cup \mathbb{B}(S_{\bar{x}})| = |\mathbb{B}(S_x)| \\ &\leq |S_x| = 2^{n-1} = \frac{1}{2} \cdot 2^n < \frac{5}{8} \cdot 2^n. \end{aligned}$$

Pero por el Lema 2.17, la desigualdad anterior es una contradicción de la maximalidad de $\mathcal{G}$, ya que para $d = \frac{n-3}{2}$, se tiene que $|B(I_3 + O_d)| = \frac{5}{8} \cdot 2^n$. Por lo tanto, $\mathcal{G}$ no puede tener vértices universales. $\square$

Recuerde que $\bar{\mathcal{G}}$ denota al complemento de la gráfica $\mathcal{G}$. Los siguientes lemas caracterizan la estructura general de $\mathcal{G}$.

Lema 2.20. *Para cualquier vértice de grado r de $\bar{\mathcal{G}}$, los biclanes de $\mathcal{G}$ cumplen la siguiente desigualdad:*

$$|B(\mathcal{G})| \leq 2^n \left(\frac{1}{2} + \frac{1}{2^r} \right).$$

Demostración. Sea $x \in \bar{\mathcal{G}}$ un vértice de grado r . Entonces en la gráfica $\mathcal{G}$, se tiene que $|N[x]| = n - r$. Como para cada conjunto $X \in S_x$, se cumple que

$N[X] \subseteq N[x]$, se sigue del Lema 2.18, que $|\mathbb{B}(S_x)| \leq 2^{|N[x]|} = 2^{n-r}$. Además, como $|\mathbb{B}(S_{\bar{x}})| \leq |S_{\bar{x}}| = 2^{n-1}$, se tiene que:

$$\begin{aligned} |\mathbb{B}(\mathcal{G})| &= |\mathbb{B}(S_{\bar{x}}) \cup \mathbb{B}(S_x)| \leq |\mathbb{B}(S_{\bar{x}})| + |\mathbb{B}(S_x)| \\ &\leq 2^{n-1} + 2^{n-r} = 2^n \left(\frac{1}{2} + \frac{1}{2^r} \right). \end{aligned}$$

□

Recordemos que $\Delta(\bar{\mathcal{G}})$ es el grado máximo de la gráfica $\bar{\mathcal{G}}$.

Lema 2.21. $\Delta(\bar{\mathcal{G}}) \leq 3$ y si $\Delta(\bar{\mathcal{G}}) = 3$, entonces $|\mathbb{B}(\mathcal{G})| \leq \frac{5}{8} \cdot 2^n$.

Demostración. Si $\bar{\mathcal{G}}$ tuviera un vértice con grado $r \geq 4$, por el Lema 2.20 se tendría que $|\mathbb{B}(\mathcal{G})| < \frac{5}{8} \cdot 2^n$, lo cual no es posible por el Lema 2.17 ($\mathcal{G}$ maximiza $|\mathbb{B}(\mathcal{G})|$), por lo tanto $\Delta(\bar{\mathcal{G}}) \leq 3$.

Por otra parte, si $\Delta(\bar{\mathcal{G}}) = 3$, entonces por el Lema 2.20, se tiene que $|\mathbb{B}(\mathcal{G})| \leq \frac{5}{8} \cdot 2^n$. □

Lema 2.22. $\bar{\mathcal{G}}$ es la unión disjunta de ciclos y trayectorias.

Demostración. Por el Lema 2.19, se tiene que $\Delta(\bar{\mathcal{G}}) \geq 2$, por lo que para mostrar que $\bar{\mathcal{G}}$ es la unión disjunta de ciclos y trayectorias, basta mostrar que $\Delta(\bar{\mathcal{G}}) \leq 2$. Para ello, suponga que $\bar{\mathcal{G}}$ tiene un vértice x de grado 3 y sea $N_{\bar{\mathcal{G}}}(x) = \{y, r, s\}$, los vecinos de x en $\bar{\mathcal{G}}$. Se mostrará que esta suposición implica que $|\mathbb{B}(\mathcal{G})| < \frac{5}{8} \cdot 2^n$, lo cual contradice al Lema 2.17.

Suponga que ninguno de los vértices en $N_{\bar{\mathcal{G}}}(x)$ es adyacente a alguno de los vértices en $\bar{\mathcal{G}} \setminus \{x, y, r, s\}$. Entonces si H es la subgráfica de $\mathcal{G}$ inducida por los vértices $N_{\bar{\mathcal{G}}}[x]$, se tiene que $\mathcal{G} = H + (\mathcal{G} \setminus H)$. Como $|H| = 4$ y $H \neq O_2$, entonces por el Teorema 2.15, se tiene que $|\mathbb{B}(H)| < |\mathbb{B}(O_2)|$. Se sigue del Teorema 2.9, la siguiente desigualdad:

$$\begin{aligned} |\mathbb{B}(\mathcal{G})| &= |\mathbb{B}(H + (\mathcal{G} \setminus H))| = |\mathbb{B}(H) \circ \mathbb{B}(\mathcal{G} \setminus H)| = |\mathbb{B}(H)| |\mathbb{B}(\mathcal{G} \setminus H)| \\ &< |\mathbb{B}(O_2)| |\mathbb{B}(\mathcal{G} \setminus H)| = |\mathbb{B}(O_2) \circ \mathbb{B}(\mathcal{G} \setminus H)| = |\mathbb{B}(O_2 + (\mathcal{G} \setminus H))|. \end{aligned}$$

Lo cual contradice la maximalidad de $\mathcal{G}$, por lo que sin pérdida de generalidad, existe un vértice $z \in \mathcal{G} \setminus \{x, y, r, s\}$, tal que z no es adyacente a y .

Puesto que la gráfica $G_0 = \mathcal{G} \setminus \{x, y\}$, tiene orden impar, por el Teorema 2.15, punto 3, se tiene que existe un vértice $t \in G_0$, tal que

$$U_0 = N_{G_0}[G_0] = N_{G_0}[G_0 \setminus \{t\}].$$

De esto se tienen los dos casos siguientes (para simplificar la notación del resto de la prueba, N sin subíndice se usará para denotar a $N_{\mathcal{G}}$):

- a) *Caso 1:* $t \neq z$. De las pruebas de los lemas 2.20 y 2.21, se tiene la siguiente desigualdad:

$$|B(\mathcal{G})| \leq |B(S_{\bar{x}})| + |B(S_x)| \leq |S_{\bar{x}}| + |B(S_x)| \leq 2^{n-1} + 2^{n-3} = \frac{5}{8} \cdot 2^n.$$

Observe que en la demostración de dicha desigualdad, se usó que $|B(S_{\bar{x}})| \leq |S_{\bar{x}}|$. Por lo que si se muestra que existen dos conjuntos diferentes S y S' de la colección $S_{\bar{x}}$, tales que $N[S] = N[S']$, entonces se tendría que $\beta(S) = \beta(S')$ y por lo tanto: $|B(S_{\bar{x}})| < |S_{\bar{x}}|$. De esto se seguirá que $|B(\mathcal{G})| < \frac{5}{8} \cdot 2^n$.

Entonces, sea $S = G_0$ y $S' = G_0 \setminus \{t\}$. Como $z \in S'$ y $|S' \cap \{r, s\}| \geq 1$, los vértices x y y no están en $N[S]$ o en $N[S']$. De esto se tiene que $N[S] = N_{G_0}[S] = U_0$ y $N[S'] = N_{G_0}[S'] = U_0$. Luego $N[S] = N[S']$, que es la contradicción deseada.

- b) *Caso 2:* $t = z$. De nuevo se usará la desigualdad del caso anterior, pero en este caso, note que en las pruebas de los lemas 2.20 y 2.21, para obtener dicha desigualdad se usó que $|B(S_x)| \leq 2^{|N[x]|} \leq 2^{n-3}$. Por lo que si se muestra que existen dos conjuntos diferentes S y S' en la colección S_x , tales que $N[S], N[S'] \in 2^{N[x]}$ y $N[S] = N[S']$, entonces se tiene que $|B(S_x)| < 2^{|N[x]|}$ y de esto se seguirá que $|B(\mathcal{G})| < \frac{5}{8} \cdot 2^n$.

Entonces, sea $S = G_0 \cup \{x\}$ y $S' = (G_0 \setminus \{t\}) \cup \{x\}$. Como $G_0 \supset \{z, r\}$, ni x ni y están en $N[G_0]$, por lo que $N[G_0] = N_{G_0}[G_0]$, luego:

$$\begin{aligned} N[S] &= N[G_0] \cap N[x] = N_{G_0}[G_0] \cap N[x] = U_0 \cap N[x] \\ &= U_0 \setminus \{r, s\}. \end{aligned}$$

Puesto que $r \in S'$, se tiene que $x \notin N[S']$ y en consecuencia

$$N[S'] = N[S'] \setminus \{x\} = N[G_0 \setminus \{t\}] \cap N(x).$$

Además, como

$$N[G_0 \setminus \{t\}] \cap G_0 = N_{G_0}[G_0 \setminus \{t\}] = U_0,$$

por lo que:

$$\begin{aligned} N[S'] &= N[G_0 \setminus \{t\}] \cap N(x) = N[G_0 \setminus \{t\}] \cap (G_0 \setminus \{r, s\}) \\ &= (N[G_0 \setminus \{t\}] \cap G_0) \setminus \{r, s\} = U_0 \setminus \{r, s\}. \end{aligned}$$

Luego $N[S] = N[S']$ y $N[S], N[S'] \in 2^{N[x]}$, por lo que $|B(\mathcal{G})| < \frac{5}{8} \cdot 2^n$, que es la contradicción deseada.

Se sigue que $\Delta(\bar{\mathcal{G}}) \leq 2$ y concluimos que $\bar{\mathcal{G}}$ es la unión disjunta de ciclos y trayectorias. $\square$

Del Lema 2.22 y de las propiedades de la suma de Zykov, la gráfica $\mathcal{G}$ es la suma de Zykov del complemento de ciclos y trayectorias:

$$\mathcal{G} = \overline{C_{m_1}} + \overline{C_{m_2}} + \cdots + \overline{C_{m_s}} + \overline{P_{n_1}} + \overline{P_{n_2}} + \cdots + \overline{P_{n_r}}.$$

Para determinar los valores de n_i y m_i , es necesario estudiar los biclanes de $\overline{C_n}$ y $\overline{P_n}$. Dejamos como un problema abierto el determinar el valor exacto de $|B(P_n)|$ y $|B(C_n)|$, sin embargo, para caracterizar $\mathcal{G}$, será suficiente dar una cota superior. Con este fin, se desarrollan el resto de los lemas de esta sección.

Lema 2.23. Sea G una gráfica con $n = |G|$ y sea $\mathcal{C}$ una colección de subconjuntos de vértices de G , tales que $N^2[S] \neq S$, para todo $S \in \mathcal{C}$. Entonces

$$|B(G)| \leq 2^n - |\mathcal{C}|.$$

Demostración. Sea $S \in \mathcal{C}$, por el teorema 2.12, $\beta(S) = (N^2[S], N[S])$ es un biclan de G . Por otra parte, por el Teorema 2.10, inciso d), se tiene que $N^4[S] = N^2[S]$ y $N^3[S] = N[S]$, luego:

$$\beta(N^2[S]) = (N^2[N^2[S]], N[N^2[S]]) = (N^4[S], N^3[S]) = (N^2[S], N[S]) = \beta(S).$$

Como $N^2[S] \neq S$, la función β no es inyectiva. Pero por el Teorema 2.15, el máximo número de biclanes (que es 2^n) solo se alcanza cuando β es inyectiva, por lo que cada conjunto de $\mathcal{C}$, disminuye en una unidad dicha cota superior. Luego $|B(G)| \leq 2^n - |\mathcal{C}|$. $\square$

El siguiente lema es útil para construir colecciones de conjuntos que satisfacen el Lema 2.23, tomando complementos de conjuntos que no puedan ser expresados como la unión de vecindades abiertas.

Lema 2.24. Sea G una gráfica y sea $Z \subseteq G$, tal que no existe $W \subseteq G$, con la propiedad de que $Z = \cup_{w \in W} N_G(w)$. Entonces para $S = \bar{Z}$, se tiene que $S \neq N_G^2[S]$.

Demostración. Note que para cualquier $v \in G$, se tiene que $N_G(v) = \overline{N_{\bar{G}}[v]}$. Entonces, para cualquier subconjunto $X \subseteq G$, se tiene la siguiente igualdad:

$$N_{\bar{G}}[X] = \bigcap_{x \in X} N_{\bar{G}}[x] = \overline{\bigcup_{x \in X} \overline{N_{\bar{G}}[x]}} = \overline{\bigcup_{x \in X} N_G(x)}.$$

Sea Z un subconjunto de vértices de G , para el que no existe subconjunto $W \subseteq G$, con la propiedad de que $Z = \bigcup_{w \in W} N_G(w)$. Suponga que $\bar{Z} = N_G^2[\bar{Z}]$, entonces de la igualdad anterior cuando $X = N_G[\bar{Z}]$, se tiene que:

$$\bar{Z} = N_G^2[Z] = N_G[N_G[\bar{Z}]] = N_G[X] = \bigcup_{x \in X} N_G(x)$$

Luego $Z = \bigcup_{x \in X} N_G(x)$, una contradicción. Por lo tanto, para $S = \bar{Z}$, se tiene que $S \neq N_G^2[S]$. $\square$

conjuntos
no-cubribles en G

colección no-cubrible

Por simplicidad, denominaremos como *conjuntos no-cubribles*, a los conjuntos de vértices de una gráfica G , que cumplan con las condiciones del Lema 2.24 (conjuntos que no pueden ser expresados como la unión de vecindades abiertas). Similarmente, si $\mathcal{C}$ es una colección de conjuntos todos no-cubribles, diremos que la *colección $\mathcal{C}$ es no-cubrible*.

El siguiente lema nos dice cómo construir conjuntos no-cubribles en G , a partir de vértices $x \in G$, cuyos vecinos también tengan vecinos (vecinos de vecinos de x). En particular, esta es una propiedad que cumplen todos los vértices de los ciclos y la mayoría de los vértices de trayectorias P_n , con $n \geq 4$.

Lema 2.25. *Sea x un vértice de una gráfica G , tal que para cada $y \in N(x)$, se tiene que $|N(y) \setminus \{x\}| \geq 1$. Sea $Z = \{z_1, \dots, z_r\}$ un conjunto de vértices de G , tal que $Z \supseteq \left(\bigcup_{y \in N(x)} N(y) \right) \setminus \{x\}$. Entonces cada conjunto en la colección $S_{x, \bar{z}_1, \dots, \bar{z}_r}$ es no-cubrible en G .*

Demostración. Sea S un conjunto de la colección $S_{x, \bar{z}_1, \dots, \bar{z}_r}$, esto es, S contiene a x y no contiene a ninguno de los vértices de Z .

Suponga que S se puede expresar como una unión de vecindades abiertas de G , entonces existe un vértice $v \in G$, tal que $N(v) \supseteq \{x\}$ y $N(v)$ no contiene a ningún vértice de Z . Como $v \in N(x)$, por hipótesis se tiene que $|N(v) \setminus \{x\}| \geq 1$ y $Z \supseteq N(v) \setminus \{x\}$. Por lo que existe un vértice $z_i \in Z \cap N(v)$, una contradicción. Luego todo conjunto de la colección $S_{x, \bar{z}_1, \dots, \bar{z}_r}$, es no-cubrible. $\square$

Por medio del Lema 2.25, se pueden obtener las siguientes cotas superiores para el número de biclanes de los complementos de caminos y trayectorias.

Lema 2.26. *El número de biclanes de los complementos de trayectorias satisface:*

1. $|B(\overline{P_2})| = 4 = 2^2$.
2. $|B(\overline{P_n})| < \frac{5}{8} \cdot 2^n$, para $n \geq 3$.

Demostración. Sea $\{0, 1, \dots, n-1\}$ el conjunto de vértices de la trayectoria P_n . Se tienen 3 casos: $2 \leq n \leq 3$, $n = 4$ y $n \geq 5$.

Caso 1: $2 \leq n \leq 3$.

Para este caso es fácil verificar manualmente las siguientes afirmaciones:

- $|B(\overline{P_2})| = 4$, con $B(\overline{P_2}) = \{ (\{0, 1\}, \emptyset), (\emptyset, \{0, 1\}), (\{0\}, \{0\}), (\{1\}, \{1\}) \}$.
- $|B(\overline{P_3})| = 4 < 5 = \frac{5}{8} \cdot 2^3$, con $B(\overline{P_3}) = \{ (\{0, 1, 2\}, \emptyset), (\emptyset, \{0, 1, 2\}), (\{0, 2\}, \{0, 2\}), (\{1\}, \{1\}) \}$.

Caso 2: $n = 4$.

Por el Lema 2.25, las siguientes son colecciones de conjuntos no-cubribles:

- $\mathcal{C}_1 = S_{0,2}^{P_4}$.
- $\mathcal{C}_2 = S_{0,1,3}^{P_4}$ es no-cubrible por $\overline{1}$ y 3.
- $\mathcal{C}_3 = S_{0,1,2,3}^{P_4}$ es no cubrible por $\overline{1}$ y 3.

Note que las colecciones de conjuntos $\mathcal{C}_1$, $\mathcal{C}_2$ y $\mathcal{C}_3$, son disjuntas por pares ya que sus conjuntos difieren en incluir/excluir los vértices 0 y 2, por lo que $\mathcal{C}_1$, $\mathcal{C}_2$ y $\mathcal{C}_3$, no pueden tener conjuntos en común. Como $|\mathcal{C}_1| = 2^{4-2} = 4$, $|\mathcal{C}_2| = 2^{4-3} = 2$ y $|\mathcal{C}_3| = 1$, se sigue de los lemas 2.23 y 2.24 que: :

$$\begin{aligned} |B(\overline{P_4})| &\leq 2^4 - |\mathcal{C}_1 \cup \mathcal{C}_2 \cup \mathcal{C}_3| = 2^4 - |\mathcal{C}_1| - |\mathcal{C}_2| - |\mathcal{C}_3| \\ &= 2^4 - 4 - 2 - 1 = 9 < 10 = 2^4 \cdot \frac{5}{8}. \end{aligned}$$

Caso 3: $n \geq 5$.

Por el Lema 2.25, las siguientes son colecciones de conjuntos no-cubribles:

- $\mathcal{C}_1 = S_{0,2}^{P_n}$.
- $\mathcal{C}_2 = S_{0,\overline{n-3},n-1}^{P_n}$ es no-cubrible por $\overline{n-3}$ y $n-1$.
- $\mathcal{C}_3 = S_{0,\overline{n-5},n-3,\overline{n-1}}^{P_n}$ es no-cubrible por $\overline{n-5}$, $n-3$ y $\overline{n-1}$.

Al igual que en el Caso 2, note que las colecciones de conjuntos $\mathcal{C}_1$, $\mathcal{C}_2$ y $\mathcal{C}_3$, son disjuntas por pares, ya que sus conjuntos difieren en incluir/excluir

los vértices 0 y $n-1$. Además, $|C_1| = 2^{n-2}$, $|C_2| = 2^{n-3}$ y $|C_3| \geq 1$. De los lemas 2.23 y 2.24, se sigue que:

$$\begin{aligned} |B(\overline{P_n})| &\leq 2^n - |C_1 \cup C_2 \cup C_3| = 2^n - |C_1| - |C_2| - |C_3| \\ &= 2^n - 2^{n-2} - 2^{n-3} - 1 = 2^n(1 - \frac{1}{4} - \frac{1}{8}) - 1 \\ &= 2^n \cdot \frac{5}{8} - 1 < 2^n \cdot \frac{5}{8}. \end{aligned}$$

□

Lema 2.27. *El número de biclanes de los complementos de ciclos satisface:*

1. $|B(\overline{C_3})| = 5 = \frac{5}{8} \cdot 2^3$.
2. $|B(\overline{C_n})| < \frac{5}{8} \cdot 2^n$, para $n \geq 4$.

Demostración. Sea $\{0, 1, \dots, n-1\}$ el conjunto de vértices del ciclo C_n . Se tienen 3 casos: $3 \leq n \leq 4$, $n = 5$ y $n \geq 6$.

Caso 1: $3 \leq n \leq 4$.

Es fácil verificar manualmente las siguientes afirmaciones:

- $|B(\overline{C_3})| = 5 = \frac{5}{8} \cdot 2^3$, con $B(\overline{C_3}) = \{ (\{0, 1, 2\}, \emptyset), (\emptyset, \{0, 1, 2\}), (\{0\}, \{0\}), (\{1\}, \{1\}), (\{2\}, \{2\}) \}$.
- $|B(\overline{C_4})| = 4 < 10 = \frac{5}{8} \cdot 2^4$, con $B(\overline{C_4}) = \{ (\{0, 1, 2, 3\}, \emptyset), (\emptyset, \{0, 1, 2, 3\}), (\{0, 2\}, \{0, 2\}), (\{1, 3\}, \{1, 3\}) \}$.

Caso 2: $n = 5$.

Por el Lema 2.25, las siguientes son colecciones de conjuntos no-cubribles:

- $\mathcal{C}_1 = S_{0, \overline{2}, \overline{3}}^{C_n}$.
- $\mathcal{C}_2 = S_{0, \overline{2}, \overline{4}}^{C_n}$.
- $\mathcal{C}_3 = S_{0, 1, \overline{2}, \overline{3}, \overline{4}}^{C_n}$ es no-cubrible por $1, \overline{3}$ y $\overline{4}$.
- $\mathcal{C}_4 = S_{0, \overline{1}, \overline{2}, 3}^{C_n}$ es no-cubrible por $\overline{0}, \overline{1}$ y 3 .
- $\mathcal{C}_5 = S_{0, \overline{1}, \overline{2}, \overline{3}, 4}^{C_n}$ es no-cubrible por $\overline{1}, \overline{2}$ y 4 .
- $\mathcal{C}_6 = S_{0, 1, \overline{2}, \overline{3}, \overline{4}}^{C_n}$ es no-cubrible por $1, \overline{3}$ y $\overline{4}$.

Note que las colecciones de conjuntos $\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_6$, son disjuntas por pares por las siguientes razones:

- $\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3$ y $\mathcal{C}_4$ difieren en incluir/excluir los vértices 0 y 2.

- $\mathcal{C}_5 \cap \mathcal{C}_4 = \emptyset$, ya que difieren en incluir/excluir el vértice 3.
- $\mathcal{C}_6 \cap \mathcal{C}_4 = \emptyset$, ya que difieren en incluir/excluir el vértice 1.
- $\mathcal{C}_5 \cap \mathcal{C}_6 = \emptyset$, ya que difieren en incluir/excluir el vértice 1.
- Para $i = 5, 6$ y $j = 1, 2, 3$, se tiene que $\mathcal{C}_i \cap \mathcal{C}_j = \emptyset$, ya que difieren en incluir/excluir los vértices 0 y 2.

Como $|\mathcal{C}_1| = |\mathcal{C}_2| = 2^{5-3} = 4$, $|\mathcal{C}_4| = 2^{5-4} = 2$ y $|\mathcal{C}_3| = |\mathcal{C}_5| = |\mathcal{C}_6| = 1$, se sigue de los lemas 2.23 y 2.24 que:

$$\begin{aligned} |B(\overline{\mathcal{C}_5})| &\leq 2^5 - |\cup_{i=1}^6 \mathcal{C}_i| = 2^5 - \sum_{i=1}^6 |\mathcal{C}_i| \leq 2^5 - (2 \cdot 4 + 2 + 3 \cdot 1) \\ &= 2^5 - 13 = 32 - 13 = 19 < 20 = \frac{5}{8} \cdot 2^5. \end{aligned}$$

Caso 3: $n \geq 6$.

Por el Lema 2.25, las siguientes son colecciones de conjuntos no-cubribles:

- $\mathcal{C}_1 = S_{0,2,n-2}^{C_n}$.
- $\mathcal{C}_2 = S_{2,0,4}^{C_n}$.
- $\mathcal{C}_3 = S_{1,3,n-1,0,2}^{C_n}$ es no-cubrible por los vértices: $1, \overline{3}$ y $\overline{n-1}$.
- $\mathcal{C}_4 = S_{1,3,n-1,0,2}^{C_n}$ es no-cubrible por los vértices: $1, \overline{3}$ y $\overline{n-1}$.
- $\mathcal{C}_5 = S_{3,1,5,0,2}^{C_n}$ es no-cubrible por los vértices: $3, \overline{1}$ y $\overline{5}$.
- $\mathcal{C}_6 = S_{3,1,5,0,2}^{C_n}$ es no-cubrible por los vértices: $3, \overline{1}$ y $\overline{5}$.
- $\mathcal{C}_7 = S_{n-1,1,n-3,0,2,n-2}^{C_n}$ es no-cubrible por los vértices: $n-1$, $\overline{1}$ y $\overline{n-3}$.

Las colecciones $\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_7$ son disjuntas por pares por las siguientes razones:

- $\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3$ y $\mathcal{C}_4$ difieren en incluir/excluir los vértices 0 y 2.
- $\mathcal{C}_5$ es disjunta por pares con $\mathcal{C}_1, \mathcal{C}_2$ y $\mathcal{C}_4$, ya que difieren en incluir/excluir los vértices 0 y 2.
- $\mathcal{C}_5 \cap \mathcal{C}_3 = \emptyset$, porque difieren en el vértice 3.
- $\mathcal{C}_6$ es disjunta por pares con $\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3$ y $\mathcal{C}_5$, ya que difieren en los vértices 0 y 2.
- $\mathcal{C}_6 \cap \mathcal{C}_4 = \emptyset$, ya que difieren en el vértice 3.
- $\mathcal{C}_7$ es disjunta por pares con $\mathcal{C}_2, \mathcal{C}_3, \mathcal{C}_4, \mathcal{C}_5$ y $\mathcal{C}_6$, porque difieren en los vértices 0 y 2.
- $\mathcal{C}_7 \cap \mathcal{C}_1 = \emptyset$, porque difieren en el vértice $n-2$.

Note que $|\mathcal{C}_1| = |\mathcal{C}_2| = 2^{n-3}$, $|\mathcal{C}_3| = |\mathcal{C}_4| = |\mathcal{C}_5| = |\mathcal{C}_6| = 2^{n-5}$ y $|\mathcal{C}_7| \geq 1$. Se sigue de los lemas 2.23 y 2.24 que:

$$\begin{aligned} |B(\overline{C_n})| &\leq 2^n - |\cup_{i=1}^7 \mathcal{C}_i| = 2^n - \sum_{i=1}^7 |\mathcal{C}_i| \leq 2^n - (2 \cdot 2^{n-3} + 4 \cdot 2^{n-5} + 1) \\ &= 2^n - 2^{n-2} - 2^{n-3} - 1 = 2^n(1 - \frac{1}{4} - \frac{1}{8}) - 1 \\ &= \frac{5}{8} \cdot 2^n - 1 < \frac{5}{8} \cdot 2^n. \end{aligned}$$

□

El teorema principal de esta sección se enuncia a continuación.

Teorema 2.28. *Sea $\mathcal{G}$ una gráfica de orden $n > 1$, que maximiza $|B(\mathcal{G})|$. Entonces, si $n = 2d$, se tiene que $\mathcal{G} \cong O_d$; en otro caso, $n = 2d + 3$ y $\mathcal{G} \cong I_3 + O_d$.*

Demostración. El caso par, $n = 2d$, es cubierto por el Teorema 2.15. Para el caso impar, $n = 2d + 3$, se tiene que $\mathcal{G} = \overline{C_{m_1}} + \overline{C_{m_2}} + \cdots + \overline{C_{m_s}} + \overline{P_{n_1}} + \overline{P_{n_2}} + \cdots + \overline{P_{n_r}}$ por el Lema 2.22 y por el Teorema 2.9, se tiene que:

$$\begin{aligned} |B(\mathcal{G})| &= |B(\overline{C_{m_1}} + \overline{C_{m_2}} + \cdots + \overline{C_{m_s}} + \overline{P_{n_1}} + \overline{P_{n_2}} + \cdots + \overline{P_{n_r}})| \\ &= |B(\overline{C_{m_1}}) \circ B(\overline{C_{m_2}}) \circ \cdots \circ B(\overline{C_{m_s}}) \circ B(\overline{P_{n_1}}) \circ B(\overline{P_{n_2}}) \circ \cdots \circ B(\overline{P_{n_r}})| \\ &= |B(\overline{C_{m_1}})| \cdot |B(\overline{C_{m_2}})| \cdots |B(\overline{C_{m_s}})| \cdot |B(\overline{P_{n_1}})| \cdot |B(\overline{P_{n_2}})| \cdots |B(\overline{P_{n_r}})|. \end{aligned}$$

Por lo que $|B(\mathcal{G})|$ es el producto de biclanes de complementos de trayectorias y ciclos. Note que $|\overline{C_3} + \overline{C_3}| = |\overline{P_2} + \overline{P_2} + \overline{P_2}|$ y por los lemas 2.26 y 2.27, se tiene que:

$$\begin{aligned} |B(\overline{C_3} + \overline{C_3})| &= |B(\overline{C_3}) \circ B(\overline{C_3})| = |B(\overline{C_3})|^2 = 5^2 = 25 \\ &< 64 = 4^3 = |B(\overline{P_2})|^3 = |B(\overline{P_2}) \circ B(\overline{P_2}) \circ B(\overline{P_2})| \\ &= |B(\overline{P_2} + \overline{P_2} + \overline{P_2})|. \end{aligned}$$

Se sigue que $\mathcal{G}$ puede alcanzar el valor máximo únicamente cuando

$$\mathcal{G} \cong \overline{C_3} + \overline{P_2} + \overline{P_2} + \cdots + \overline{P_2} = I_3 + I_2 + I_2 + \cdots + I_2 = I_3 + O_d.$$

□

Note que de acuerdo al Teorema 2.28, la gráfica G que maximiza $|B(G)|$, es la misma gráfica $\mathcal{G}_n$, que se definió en 2.3.

Neumann-Lara demostró en [49], que la gráfica de clanes del octaedro d -dimensional O_d , es un octaedro r -dimensional, con $r = 2^{d-1}$, es decir:

$$K(O_d) \cong O_r = O_{2^{d-1}}.$$

Por el Teorema 2.8, se tiene que $K^2(S(G)) \cong B(K(G))$ y como $O_d = S(O_{d-1})$, se sigue que:

$$K^2(O_d) = K^2(S(O_{d-1})) \cong B(K(O_{d-1})) \cong B(O_{2^{d-2}}).$$

Este resultado, junto con el Teorema 2.28, **sugieren** que si G tiene un orden par, las gráficas que maximizan $|K^2(G)|$ son las mismas que maximizan $|B(G)|$ (excepto cuando $n = 4$, en este caso, I_4 también maximiza $|K^2(G)|$). Ciertamente, se tiene la siguiente cota inferior:

$$|K^2(G)| \geq |K^2(O_d)| = |B(O_{2^{d-2}})| = 2^{2^{d-1}} = \sqrt{2}^{\sqrt{2}^n}.$$

Para el caso impar, la evidencia experimental que se mencionó con anterioridad, **sugiere** que las gráficas que maximizan $|K^2(G)|$, también son las mismas que maximizan $|B(G)|$ (excepto para algunas excepciones mencionadas antes en la Sección 2.1).

Si bien, el problema de determinar cuál es la gráfica que maximiza $|K^2(G)|$, sigue siendo un problema abierto, en la siguiente sección presentamos algunos resultados y conjeturas, que creemos podrían ser útiles para resolver este problema.

2.3 RESULTADOS ADICIONALES Y CONJETURAS

Lema 2.29. $\overline{G \circ I_2}$ es una gráfica bipartita.

Demostración. Sea $H = G \circ I_2$ y sea $I_2 = \{x, y\}$. Sea $V(H) = \{(v, z) : v \in G \text{ y } z \in I_2\}$, el conjunto de vértices de la gráfica H . Defina los conjuntos $X = \{(v, x) : v \in G\}$ y $Y = \{(v, y) : v \in G\}$. Entonces $X \cap Y = \emptyset$ y $V(H) = X \cup Y$. Luego X y Y forman una partición de $V(H) = V(\overline{H})$.

Sean (v, x) y (v', x) vértices del conjunto X , con $v \neq v'$. Por la definición del producto bola, se tiene que $(v, x) \simeq (v', x)$, en H . De esto se tiene que en $\overline{H}$, los vértices (v, x) y (v', x) no son adyacentes. Por lo tanto, X es un conjunto independiente en $\overline{H}$. Un argumento similar muestra que Y también es un conjunto independiente en $\overline{H}$. Como X y Y son conjuntos independientes en $\overline{H}$, que forman una partición de $V(\overline{H})$, entonces $\overline{H}$ es una gráfica bipartita. $\square$

Lema 2.30. Si $\overline{G}$ es bipartita, entonces $G \circ I_2 \cong G + G$

Demostración. Como $\overline{G}$ es una gráfica bipartita, existen conjuntos de vértices independientes X y Y de $\overline{G}$ (que son completas en G), tales que X y Y son una partición de los vértices de G . Sea $I_2 = \{z_1, z_2\}$ y sea $V(G \circ I_2) = \{(v, z) : v \in G \text{ y } z \in I_2\}$, el conjunto de vértices de la gráfica $G \circ I_2$.

Defina los conjuntos $X_1 = \{(x, z_1) : x \in X\}$, $Y_1 = \{(y, z_1) : y \in Y\}$ y $X_2 = \{(x, z_2) : x \in X\}$ y $Y_2 = \{(y, z_2) : y \in Y\}$. Note que los conjuntos X_1, Y_1, X_2 y Y_2 son una partición de $V(G \circ I_2)$. Sean las gráficas inducidas $G_1 = X_1 \cup Y_2$ y $G_2 = X_2 \cup Y_1$, entonces $V(G \circ I_2) = V(G_1) \cup V(G_2)$.

Defina la función $\phi_1 : G_1 \rightarrow G$, como $\phi_1((v, z)) = v$. Se mostrará que ϕ es un isomorfismo de gráficas. Para mostrar que ϕ_1 es biyectiva, sea $v \in G$, entonces $v \in X$ o $v \in Y$. Si $v \in X$, entonces $(v, z_1) \in X_1 \subset G_1$ y $\phi_1(v, z_1) = v$ y si $v \in Y$, entonces $(v, z_2) \in Y_2 \subset G_1$ y $\phi_1(v, z_2) = v$, luego ϕ_1 es sobre. Por otra parte, si $\phi_1(v, z) = \phi_1(v', z')$, entonces $\phi_1(v, z) = v$ y $\phi_1(v', z') = v'$, luego $v = v'$ y ϕ_1 es inyectiva.

Para mostrar que ϕ_1 preserva adyacencias, sean $(v, z), (v', z') \in G_1$, tales que $(v, z) \simeq (v', z')$, se tienen dos casos:

- *Caso 1:* $z = z'$. Entonces $z = z_1$ o $z = z_2$, si $z = z_1$, entonces $v, v' \in X$ y como X es una completa en G , se tiene que $v \simeq v'$, como $\phi_1(v, z) = v$ y $\phi_1(v', z) = v'$, entonces $\phi_1(v, z) \simeq \phi_1(v', z)$. De manera similar, para $z = z_2$, se tiene que $v, v' \in Y$ y como Y es una completa en G , $\phi_1(v, z) \simeq \phi_1(v', z)$.
- *Caso 2:* $z \neq z'$, entonces sin pérdida de generalidad $v \in X$ y $v' \in Y$. Por hipótesis $(v, z) \simeq (v', z')$ y como $z \neq z'$, por definición de producto bola esto implica que $v \simeq v'$. $\phi_1(v, z) = v \simeq \phi_1(v', z) = v'$.

De lo anterior se tiene que ϕ_1 es un isomorfismo de gráficas. Un argumento similar muestra que la función $\phi_2 : G_2 \rightarrow G$, como $\phi_2((v, z)) = v$, también es un isomorfismo de gráficas. Luego $G_1 \cong G$ y $G_2 \cong G$.

Sea $(x, z_1) \in G_1$, con $x \in X$ y sea $(v, z) \in G_2$. Si $z = z_1$, por la definición del producto bola se tiene que $(x, z_1) \simeq (v, z)$. Si $z = z_2$, entonces $v \in X$ y como X es una completa en G , $x \simeq v$, luego $(x, z_1) \simeq (v, z)$. Por lo tanto (x, z_1) es adyacente a todo vértice de G_2 . Un argumento similar muestra que $(y, z_2) \in G_1$, es adyacente a todo vértice de G_2 . Luego $G \circ I_2 \cong G_1 + G_2 \cong G + G$. $\square$

Para el resto de esta sección y para un entero $d \geq 1$, usaremos la notación: $G^d = G \circ G \circ \dots \circ G$ (d veces). En particular, $I_2^d = I_2 \circ I_2 \circ \dots \circ I_2$ (d veces)

y $K(G)^2 = K(G) \circ K(G)$. Note que $K(G)^2$, no es lo mismo que $K^2(G)$, esta última denota la segunda gráfica de clanes de G . Esta notación también cumple las reglas: $G^a \circ G^b = G^{a+b}$ y $(G^a)^b = G^{ab}$.

Teorema 2.31. Si $\overline{G}$ es bipartita, entonces $K(G \circ I_2) = K(G)^2$

Demostración. Por el Lema 2.30 y por las propiedades del operador de clanes K , con la suma de Zykov, se tiene que:

$$K(G \circ I_2) = K(G + G) = K(G) \circ K(G) = K(G)^2$$

□

Teorema 2.32. Para $d \geq 1$, $K(G \circ I_2^d) = K(G \circ I_2)^{2^{d-1}}$.

Demostración. La demostración será por inducción sobre d . Para $d = 1$, la prueba es trivial. Para $d \geq 2$, por el Lema 2.29, la gráfica:

$$H = \overline{G \circ I_2^{d-1}}$$

es una gráfica bipartita y por el Teorema 2.31, se tiene que $K((G \circ I_2^{d-1}) \circ I_2) = K(G \circ I_2^{d-1})^2$. Por la hipótesis de inducción $K(G \circ I_2^{d-1}) = K(G)^{2^{d-1}}$, luego:

$$K(G \circ I_2^d) = K((G \circ I_2^{d-1}) \circ I_2) = K(G \circ I_2^{d-1})^2 = (K(G)^{2^{d-1}})^2 = K(G)^{2^d}$$

□

Teorema 2.33. Si $c \geq 1$, entonces

$$K^2(c \cdot I_3 + d \cdot I_2) = K(I_3^c \circ I_2)^{2^{d-1}}.$$

Demostración. Por las propiedades del operador de clanes con la suma de Zykov, se tiene que $K(c \cdot I_3 + d \cdot I_2) = I_3^c \circ I_2^d$. Luego, por el Teorema 2.32:

$$K^2(c \cdot I_3 + d \cdot I_2) = K(I_3^c \circ I_2^d) = K(I_3^c \circ I_2)^{2^{d-1}}.$$

□

El teorema 2.33, es útil para calcular el número de clanes de la gráfica $\mathcal{G}_n$ (Definición 2.3). Este resultado se enuncia en el siguiente corolario:

Corolario 2.34. Sea $\mathcal{G}_n$ la gráfica definida en 2.3, entonces:

$$|K^2(\mathcal{G}_n)| = \begin{cases} \sqrt{2}^{\sqrt{2}^n}, & \text{si } n \text{ es par,} \\ \sqrt{5}^{\sqrt{2}^{n-3}}, & \text{si } n \text{ es impar.} \end{cases}$$

Demostración. Si n es par, entonces $\mathcal{G}_n = O_{\frac{n}{2}}$. Como $K(O_d) \cong O_{2^{d-1}}$ y $|K(O_d)| = 2^d$, se sigue que:

$$|K^2(\mathcal{G}_n)| = |K^2(O_{\frac{n}{2}})| = |K(O_{2^{\frac{n}{2}-1})| = 2^{2^{\frac{n}{2}-1}} = \sqrt{2}^{\sqrt{2}^n}$$

Si n es impar, entonces $\mathcal{G}_n = I_3 + O_{\frac{n}{2}}$. Es fácil verificar que $|K(I_3 \circ I_2)| = 5$. Entonces, para $d = \frac{n-3}{2}$, del Teorema 2.32, se sigue que:

$$\begin{aligned} |K^2(\mathcal{G}_n)| &= |K^2(I_3 + O_d)| = |K^2(I_3 + d \cdot I_2)| = |K(I_3 \circ I_2)^{2^{d-1}}| \\ &= |K(I_3 \circ I_2)|^{2^{d-1}} = 5^{2^{d-1}} = 5^{2^{\frac{n-3}{2}-1}} = \sqrt{5}^{\sqrt{2}^{n-3}}. \end{aligned}$$

□

Lema 2.35. Sea $f(n)$, la función en los enteros dada por:

$$f(n) = \begin{cases} \sqrt{2}^{\sqrt{2}^n}, & \text{si } n \text{ es par,} \\ \sqrt{5}^{\sqrt{2}^{n-3}}, & \text{si } n \text{ es impar.} \end{cases}$$

Entonces $f(n) = f(n-2)^2$.

Demostración. Observe que para cualquier entero a y $x > 0$, se cumple que:

$$\left(x^{\sqrt{2}^{a-2}}\right)^2 = x^{2 \cdot \sqrt{2}^{a-2}} = x^{\sqrt{2}^2 \cdot \sqrt{2}^{a-2}} = x^{\sqrt{2}^a}.$$

Por lo que si n es par, con $a = n$ y $x = \sqrt{2}$, se obtiene $f(n) = f(n-2)^2$. De manera similar, si n es impar, con $a = n-3$ y $x = \sqrt{5}$, se obtiene $f(n) = f(n-2)^2$. □

Tenemos la siguiente conjetura que hemos formulado basándonos en los resultados experimentales mencionados en la Sección 2 y en los resultados presentados en este capítulo:

Conjetura 2.36. *Para cualquier gráfica G , con $n \geq 6$ vértices, se cumple que:*

$$|K^2(G)| \leq f(n) = \begin{cases} \sqrt{2}^{\sqrt{2}^n}, & \text{si } n \text{ es par,} \\ \sqrt{5}^{\sqrt{2}^{n-3}}, & \text{si } n \text{ es impar.} \end{cases}$$

La igualdad se cumple si $G = \mathcal{G}_n$ (gráfica de la definición 2.3).

Si se pudiera probar que $K^2(G)$ se maximiza, cuando $G = H + I_2$ con $\overline{K(H)}$ bipartita, o también, cuando $G = T + I_2 + I_2$ (por el lema 2.29, $\overline{K(T + I_2)}$ es bipartita), por el teorema 2.31, se tendría que $|K^2(G)| = |K(H)|^2$ o $|K^2(G)| = |K(T + I_2)|^2$. Entonces usando inducción sobre n y por el lema 2.35, se podría demostrar la conjetura 2.36.

ALGORITMOS GENÉTICOS

Uno de los problemas que se investigaron en este trabajo doctoral, fue el determinar si existe una gráfica G , cuyas gráficas iteradas de clanes $K^n(G)$, tienen crecimiento exponencial (i.e. $|K^n(G)| = \Theta(\alpha^n)$, para $\alpha > 1$). Tenemos la conjetura de que la gráfica circulante $C_n(1,3,6,7,8)$, con $n \geq 25$, tiene dicho comportamiento: en la sección 3.5, se presentan los resultados que llevaron a plantear dicha conjetura. La gráfica $C_{25}(1,3,6,7,8)$, fue la que inicialmente se encontró en este trabajo por medio de una heurística basada en algoritmos genéticos, la cual se describe a detalle en este capítulo.

Los algoritmos genéticos pertenecen a un grupo de metaheurísticas denominadas *algoritmos evolutivos*, las cuales se usan para resolver problemas de *búsqueda y optimización*, por medio de técnicas que están inspiradas en algunos de los mecanismos de la *evolución biológica*, como lo son: selección natural, reproducción, recombinación y mutación [45].

*algoritmos
evolutivos
evolución biológica*

Uno de los primeros trabajos en los que se investigó la posibilidad de simular la evolución biológica mediante el uso de computadoras y usar dicho mecanismo para resolver problemas de optimización, fue el de Barricelli, en 1954 [3]. Posteriormente, durante las décadas de 1950 y 1960, los trabajos de Fraser [17] y Friedberg [21, 22], contribuyeron al desarrollo de las bases de los algoritmos evolutivos. Sin embargo, no fue hasta el año de 1975, que en el trabajo de Holland [35], se presenta la terminología y heurísticas de lo que se conoce actualmente como *algoritmos genéticos (AGs)*.

*Algoritmos
Genéticos (AGs)*

Son diversos los problemas de optimización en los que se han utilizado los AGs: se han aplicado en problemas relacionados con el problema de la persona viajante [44], en alineamiento de secuencias en bioinformática [51], en detección de errores en programas de computadora [37] e incluso para el desarrollo de antenas [36]. En particular, en este trabajo se han usado AGs para encontrar una gráfica cuyas gráficas iteradas de clanes, tienen crecimiento exponencial.

En principio, se puede buscar una gráfica que cumpla con características que son de interés (como puede ser que tenga crecimiento exponencial con el operador de clanes), con un enfoque de fuerza bruta: esto es, dado un entero n , se revisan todas las gráficas de n vértices para deter-

minar si hay alguna gráfica, que experimentalmente cumple con las características deseadas. Sin embargo, en general no es factible realizar una búsqueda por fuerza bruta en un tiempo razonable. Por ejemplo, existen 64001015704527557894928 gráficas de 16 vértices (salvo isomorfismos) [59]. Aún usando una de las supercomputadoras más rápidas del mundo (que teóricamente podría realizar $2 \cdot 10^{18}$ operaciones por segundo [58]) y si se supone que se puede procesar una gráfica, en 2000 operaciones (una estimación extremadamente optimista), llevaría más 2 años procesar todas las gráficas de 16 vértices.

Puesto que una búsqueda por fuerza bruta no es factible, es necesario usar alguna heurística que permita hacer una búsqueda de una forma más dirigida, de tal forma que permita encontrar una gráfica con las características deseadas en un tiempo razonable. Para los fines de esta investigación, una heurística basada en algoritmos genéticos dio resultados satisfactorios. Sin embargo, otro tipo de heurística que ya no se usó pero que quizás pueda ser de utilidad para los fines de este trabajo, es la del algoritmo de recocido simulado [11].

Para implementar un algoritmo genético, primero se debe elegir una *codificación del problema* (con su respectiva *decodificación*), luego se tiene que definir una *función de aptitud* y finalmente se aplican los *pasos generales* de la heurística que conforma a un algoritmo genético. Cada uno de estos tópicos y cómo se aplicaron en este trabajo, se explican a detalle en las siguientes secciones.

3.1 CODIFICACIÓN Y DECODIFICACIÓN EN UN ALGORITMO GENÉTICO

espacio de búsqueda El *espacio de búsqueda* de un problema de optimización, es el conjunto de elementos que satisfacen las restricciones del problema [4]. A los elementos del espacio de búsqueda se les conoce como *soluciones candidatas* (una solución candidata no necesariamente es una solución óptima). Para poder aplicar un algoritmo genético a un problema de optimización, primero se tiene que elegir una *codificación* para las soluciones candidatas.

genes La codificación en los AGs es análoga a la codificación de información genética que ocurre en los seres vivos y que se describe brevemente a continuación. En los organismos, los *genes* contienen información para construir *proteínas*, las cuales son esenciales para su funcionamiento. Cada gen tiene variantes conocidas como *alelos*, que determinan características físicas y bioquímicas del organismo [64]. Por ejemplo, el gen LCT codifica a la

enzima lactasa (una proteína) y las variantes de este gen (alelos), determinan que una persona pueda ser intolerante/tolerante a la lactosa [62]. La colección de información presente en los genes se denomina *información genética*.

información genética

De manera similar, en los AGs los "organismos" son las soluciones candidatas de un problema de optimización y los "genes" de una solución candidata son las características de dicha solución. Para representar computacionalmente la "información genética" de cada solución, se elige una *codificación*; que por lo general es una cadena de unos y ceros (codificación binaria).

codificación

Dado el conjunto de gráficas de n vértices (simples y no dirigidas), un problema en cuestión es el buscar en dicho conjunto una familia de gráficas con ciertas propiedades de interés. Desde el enfoque de los AGs, se puede pensar que en este problema las gráficas son "organismos" y sus "características" son sus adyacencias. La codificación que se utilizó en este trabajo es la siguiente (véase figura 1): se obtiene la matriz de adyacencia de la gráfica y se toma cada renglón arriba de la diagonal principal; cada renglón se concatena yendo de arriba hacia abajo, para obtener una cadena de ceros y unos. Observe que la longitud L de la cadena que codifica a una gráfica de n vértices, esta dada por: $L = \frac{n(n-1)}{2}$.

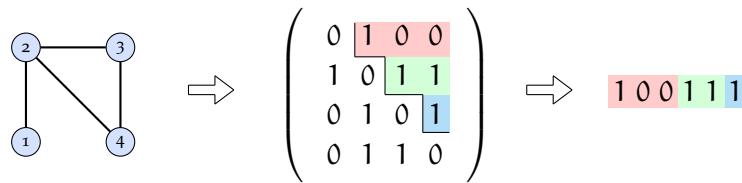


Figura 1: Codificación de una gráfica en una cadena binaria.

El proceso inverso de la codificación es la *decodificación*. En nuestro caso, la decodificación obtiene la gráfica G a partir de su codificación binaria mediante los siguientes pasos (véase la figura 2):

decodificación

1. Si L es la longitud de la codificación, el total de vértices n de G se calcula con $n = \frac{1+\sqrt{8L+1}}{2}$. Por ejemplo, la cadena 100111, tiene longitud $L = 6$. Por lo que:

$$n = \frac{1 + \sqrt{8L + 1}}{2} = \frac{1 + \sqrt{8(6) + 1}}{2} = \frac{1 + \sqrt{49}}{2} = 4.$$

2. Se lee de izquierda a derecha la cadena: los primeros $n - 1$ dígitos, forman el primer renglón arriba de la diagonal principal. Los $n - 2$ dígitos siguientes, forman el segundo renglón y así sucesivamente hasta obtener la matriz triangular superior (excluyendo la diagonal principal, que sólo contiene ceros). El resto de la matriz (la matriz

triangular inferior), se obtiene transponiendo la matriz triangular superior.

3. Se usa la matriz de adyacencias para construir a la gráfica.

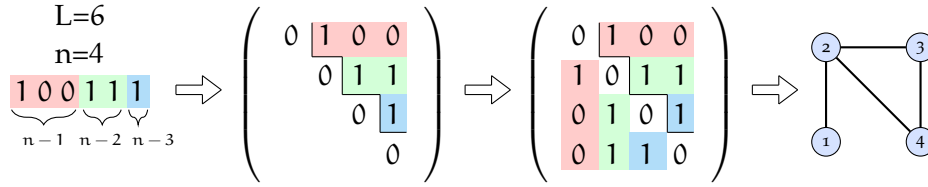


Figura 2: Decodificación de una cadena binaria que representa una gráfica.

3.2 FUNCIÓN DE APTITUD

selección natural En la evolución biológica, la *selección natural* es el proceso en el que un organismo con mayor *aptitud* para adaptarse a un ambiente, tiene más probabilidades de *reproducirse* y dejar descendencia [45]. De manera similar, en los AGs a cada solución candidata se le asigna un número real positivo que representa su *aptitud*: mientras mayor sea dicha aptitud, más probabilidades hay de que la solución candidata sea seleccionada para reproducción (este paso de los AGs se explica más adelante, en la sección 3.3.2).

aptitud

función de aptitud f_A Si S es el espacio de búsqueda del problema de optimización y R^+ es el conjunto de números reales positivos, la asignación de la aptitud en un algoritmo genético se hace mediante la *función de aptitud* $f_A : S \rightarrow R^+$. La definición de la *función de aptitud* depende del problema de optimización, pero por lo general se suele definir de tal forma que la función f_A mide que tan óptima es una solución candidata.

Para especificar la función de aptitud que se usó en esta investigación, conviene primero definir la siguiente función.

Función de crecimiento **Definición 3.1** (Función de crecimiento de una gráfica). Sea G una gráfica y sea $K^n(G)$, su n -ésima gráfica iterada de clanes. Su función de crecimiento $g_G(n)$, se define como:

$$g_G(n) = |K^n(G)|.$$

En este trabajo se tiene interés en encontrar gráficas G , para las que su función $g_G(n)$ tiene crecimiento exponencial (si es que existen). Es por ello que la función de aptitud f_A se ha propuesto como una medida de que tan óptimo es aproximar a la función $g_G(n)$, con el modelo de regresión exponencial:

$$y_e(n) = A \cdot B^n, \quad (3.1)$$

donde A y B son constantes a determinar. En principio, en el modelo 3.1 se podría agregar una constante C , para así tener $A \cdot B^n + C$ y dar más precisión al modelo. Sin embargo, excluir dicha constante simplifica el cálculo de la función de aptitud, ya que esto permite transformar el modelo 3.1, en un modelo de regresión lineal, que para nuestros fines este enfoque fue suficiente para obtener resultados satisfactorios.

Para obtener un modelo de regresión lineal de 3.1, se aplica logaritmo natural para obtener:

$$\text{Ln}(y_e(n)) = \text{Ln}(A \cdot B^n) = \text{Ln}(A) + \text{Ln}(B) \cdot n.$$

Sustituyendo en la ecuación anterior $y(n) = \text{Ln}(y_e(n))$, $a = \text{Ln}(A)$ y $b = \text{Ln}(B)$, se obtiene la siguiente ecuación lineal:

$$y(n) = a + bn. \quad (3.2)$$

Bajo este nuevo enfoque, f_A debería medir que tan óptimo es aproximar a $\text{Ln}(g_G(n))$, con el modelo de regresión lineal $y(n)$. El valor de las constantes a y b , se calculan por medio del método de *mínimos cuadrados* [13].

Para el resto de esta sección conviene tener presentes las siguientes definiciones.

Definición 3.2 (Promedio, varianza y covarianza). Sean $X = (x_1, x_2, \dots, x_n)$ y $Y = (y_1, y_2, \dots, y_n)$, vectores de n entradas, el promedio μ_X , la varianza $\text{Var}(X)$ y la covarianza $\text{Cov}(X, Y)$, se definen como:

promedio, varianza y covarianza

- $\mu_X = \frac{\sum_{i=1}^n x_i}{n}.$
- $\text{Var}(X) = \frac{\sum_{i=1}^n (x_i - \mu_X)^2}{n}.$
- $\text{Cov}(X, Y) = \frac{\sum_{i=1}^n (x_i - \mu_X)(y_i - \mu_Y)}{n}.$

Las fórmulas para a y b , de la regresión 3.2, se muestran en el siguiente lema (por ser un resultado ampliamente conocido en la literatura, se omite la demostración):

Lema 3.3. Sean X y Y , vectores de N entradas. Sea $y(n) = a + bn$, la regresión lineal por mínimos cuadrados que se usa para aproximar la relación entre X y Y . Entonces las constantes a y b , están dadas por las siguientes fórmulas:

$$b = \frac{\text{Cov}(X, Y)}{\text{Var}(X)} \quad y \quad a = \mu_Y - b \cdot \mu_X.$$

Donde Var , Cov , μ_Y y μ_X , son la varianza, covarianza, promedio de X y promedio de Y , respectivamente.

Como en este trabajo la regresión lineal del lema 3.3, se usa para aproximar el logaritmo de $g_G(n)$, resulta útil tener la siguiente definición.

Definición 3.4 (Vector de crecimiento de una gráfica). *Sea G una gráfica con función de crecimiento $g_G(n)$. Dado un entero $N \geq 1$, el vector de crecimiento Y_G de la gráfica G , se define como:*

$$Y_G = (\text{Ln}(g_G(0)), \text{Ln}(g_G(1)), \dots, \text{Ln}(g_G(N))).$$

Note que en la definición 3.4, N es el entero más grande para el que se desea calcular el n -ésimo término $\text{Ln}(g_G(n))$, del vector Y_G . Para los experimentos computacionales de este trabajo, se usó $N \leq 7$, ya que valores más grandes ocasionan un tiempo de cálculo muy extenso.

Para asociar a cada gráfica G con una aptitud, primero se tiene que calcular su vector de crecimiento Y_G . Posteriormente se debe medir qué tan óptimo es aproximar los puntos de Y_G , con la regresión lineal $y(n)$: esto se puede hacer por medio del *coeficiente de correlación* [12], cuya definición se ha adaptado para el contexto de este trabajo y se describe a continuación.

Definición 3.5 (Coeficiente de correlación). *Sea G una gráfica con vector de crecimiento Y_G y sea el vector $X_G = (0, 1, \dots, |Y_G| - 1)$. El coeficiente de correlación ρ_G , de la gráfica G , se define como:*

$$\rho_G = \frac{\text{Cov}(X_G, Y_G)}{\sqrt{\text{Var}(X_G) \cdot \text{Var}(Y_G)}}.$$

Donde Var y Cov , son la varianza y covarianza respectivamente.

Observación 3.6. *En la definición 3.5, el coeficiente de correlación no está definido, si $\text{Var}(X_G) = 0$ o si $\text{Var}(Y_G) = 0$.*

El cuadrado del coeficiente de correlación, tiene la propiedad de que su valor tiende a 1, mientras mejor sea la aproximación de un conjunto de puntos con una regresión lineal y tiende a 0, en caso contrario [12]. Esta propiedad convierte al coeficiente de correlación en una parte esencial para poder definir la función de aptitud de este trabajo.

Dada una gráfica G , en principio se podría usar a ρ_G^2 , como una función de aptitud, pero se debe considerar la siguiente problemática: existen gráficas cuya función $g_G(n)$, puede tener un crecimiento mucho mayor al que se está buscando (exponencial para los fines de este trabajo). Por ejemplo,

existen gráficas para las que el orden de sus gráficas iteradas de clanes tiene crecimiento súper exponencial [40], en estos casos resulta computacionalmente imposible calcular $g_G(n)$, para valores de n tan pequeños como $n = 4$. Por lo que para este tipo de gráficas, a partir de cierta n , ya no se puede calcular el vector de crecimiento Y_G .

Para evitar problemas derivados de que para una gráfica G , su función $g_G(n)$ pueda crecer demasiado, se puede hacer lo siguiente: buscar los clanes de la gráfica $K^n(G)$, llevando la cuenta del total de clanes encontrados al momento, si este número excede cierto umbral, detener la búsqueda; si nunca se excede el umbral, se procede a calcular $K^n(G)$ y con ello $g_G(n)$. De esta forma, para calcular el vector de crecimiento Y_G , se usan los valores de la función $g_G(n)$, que no excedan el umbral de crecimiento. El pseudocódigo que se usó en este trabajo, para calcular Y_G , se muestra en el algoritmo 1.

Algoritmo 1: cálculo del vector de crecimiento Y_G , de una gráfica G .

Entradas: una gráfica G ; el número máximo N de gráficas iteradas de clanes a calcular (es decir, como máximo se calcula hasta $K^N(G)$); un número M que sirve para detener el algoritmo, en caso de que $|K^n(G)| > M \cdot |K^{n-1}(G)|$, con $n \geq 1$.

Salida: una lista Y_G , que representa el vector de crecimiento de G , sujeto a las restricciones de los parámetros N y M .

```

1  Función VectorDeCrecimiento( $G, N, M$ )
2       $Y_G = [ \text{Ln}(|G|) ]$ 
3       $H = G$ 
4      para cada  $n \in \{1, 2, \dots, N\}$  hacer
5           $\text{maxCli} = M \cdot |H|$ 
6           $\text{total} = \text{NumberOfCliques}(H, \text{maxCli})$ 
7          si  $\text{total} \geq \text{maxCli}$  entonces
8              regresar  $Y_G$ 
9          en otro caso
10              $K(H) = \text{CliqueGraph}(H)$ 
11              $Y_G[n + 1] = \text{Ln}(|K(H)|)$ 
12              $H = K(H)$ 
13         fin
14     fin
15     regresar  $Y_G$ 
16 fin

```

El número M que recibe la función VectorDeCrecimiento, del algoritmo 1, sirve como umbral para el crecimiento de la función $g_G(n)$. Para esta in-

vestigación se usó $M = 4$, el cual fue un valor suficiente para obtener resultados satisfactorios.

funciones
NumberOfCliques y
CliqueGraph de
YAGS

Las funciones *NumberOfCliques* y *CliqueGraph* de YAGS, que están en las líneas 6 y 10, del algoritmo 1, sirven para contar clanes y calcular gráficas iteradas de clanes, respectivamente. Ambas funciones usan el algoritmo de Bron-Kerbosh [6], para calcular clanes.

La variable H , en el algoritmo 1, contiene la última gráfica $K^n(G)$ que se ha calculado. Por otra parte, el parámetro maxCli se usa como cota superior para el orden de la gráfica $K^{n+1}(G)$. El valor de maxCli se calcula mediante la siguiente formula (algoritmo 1, línea 5):

$$\text{maxCli} = M \cdot |H|.$$

La función *NumberOfCliques* (algoritmo 1, línea 6), de la biblioteca YAGS, cuenta los clanes de la gráfica H , con la siguiente peculiaridad: la función busca los clanes y cada vez que encuentra un clan, incrementa un contador; si dicho contador excede el umbral maxCli , la función detiene la búsqueda de clanes. Dependiendo de esto ocurren dos casos:

- **Caso 1:** el umbral maxCli se alcanza (algoritmo 1, línea 7). En este caso la función *VectorDeCrecimiento* termina y regresa a Y_G (algoritmo 1, línea 8). Por lo que en este caso Y_G cumple que:

$$|Y_G| < N + 1.$$

- **Caso 2:** el umbral maxCli no se alcanza (algoritmo 1, línea 9). En este caso se calcula la gráfica de clanes $K(H)$ de la gráfica H (algoritmo 1, línea 10). Posteriormente, el valor $\text{Ln}(|K(H)|)$ se agrega a Y_G y se actualiza H (algoritmo 1, líneas 11 y 12). Si para $0 \leq n \leq N$, nunca se sobrepasa el valor maxCli , entonces al terminar el algoritmo 1 se cumple que:

$$|Y_G| = N + 1.$$

Note que el tamaño de Y_G en el algoritmo 1, cumple que: $|Y_G| \leq N + 1$.

El siguiente lema muestra que si se calcula Y_G , con el algoritmo 1, entonces los términos de Y_G , están acotados por una función lineal.

Lema 3.7. Sea G una gráfica y sea Y_G el vector de crecimiento que se obtiene del algoritmo 1, al recibir la gráfica G y los parámetros N y M . Para cualquier elemento $\text{Ln}(|K^n(G)|) \in Y_G$, se cumple que:

$$\text{Ln}(|K^n(G)|) \leq n \cdot \text{Ln}(M) + \text{Ln}(|G|).$$

Demostración. La prueba se hará por inducción sobre n . Cuando $n = 0$, se tiene que:

$$\text{Ln}(|K^0(G)|) = \text{Ln}(|G|) = 0 \cdot \text{Ln}(M) + \text{Ln}(|G|)$$

Suponga que la afirmación es cierta para un entero $j < N$. Si $\text{Ln}(|K^{j+1}(G)|)$ es un valor de Y_G , entonces dicho valor fue agregado a Y_G , en la línea 11, del algoritmo 1: por lo que en la línea 10 (una línea antes), $K(H) = K^j(G)$ y por las líneas 5 y 6 se tiene que:

$$|K^{j+1}(G)| < \text{maxCli} = M \cdot |K(H)| = M \cdot |K^j(G)|$$

Al aplicar logaritmo a la desigualdad anterior queda:

$$\text{Ln}(|K^{j+1}(G)|) < \text{Ln}(M \cdot |K^j(G)|) = \text{Ln}(M) + \text{Ln}(|K^j(G)|)$$

Por el paso inductivo $\text{Ln}(|K^j(G)|) \leq j \cdot \text{Ln}(M) + \text{Ln}(|G|)$, al sustituir en la desigualdad anterior se obtiene:

$$\begin{aligned} \text{Ln}(|K^{j+1}(G)|) &< \text{Ln}(M) + \text{Ln}(|K^j(G)|) = \text{Ln}(M) + j \cdot \text{Ln}(M) + \text{Ln}(|G|) \\ &= (j+1)\text{Ln}(M) + \text{Ln}(|G|) \end{aligned}$$

□

Por el lema 3.7, si el vector de crecimiento Y_G que se obtiene con el algoritmo 1, tiene cardinalidad $|Y_G| < N + 1$, entonces la gráfica G tiene $N + 1 - |Y_G|$ gráficas iteradas de clanes que exceden el umbral de crecimiento deseado: por lo que la función de aptitud debería dar un valor mayor, para aquellas gráficas para las que $|Y_G| = N + 1$ y un valor menor, si $|Y_G| < N + 1$. Es por ello que en este trabajo la definición final para la función de aptitud es la siguiente:

Definición 3.8 (Función de aptitud de crecimiento exponencial). *Sea G una gráfica con vector de crecimiento Y_G , que se obtuvo mediante el algoritmo 1, con los parámetros N y M . Sea el vector $X_G = (0, 1, \dots, |Y_G| - 1)$ y sea ρ_G , el coeficiente de correlación de G . La función de aptitud de crecimiento exponencial f_A , de la gráfica G , se define como:*

$$f_A(G) = \begin{cases} \rho_G^2 \cdot \left(\frac{|Y_G|}{N+1} \right), & \text{si } \text{Var}(Y_G) \neq 0 \text{ y } \text{Var}(X_G) \neq 0, \\ 0, & \text{en otro caso.} \end{cases}$$

Donde $\text{Var}(Y_G)$ y $\text{Var}(X_G)$, son la varianza de Y_G y X_G , respectivamente.

Note que en la definición 3.8, si G es una gráfica en la que su vector Y_G , tiene todas sus entradas iguales, entonces $\text{Var}(Y_G) = 0$ y la función f_A le asigna la menor aptitud posible, que es cero. Por otra parte, el cociente

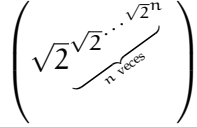
NÚMERO	GRÁFICA G	CRECIMIENTO $g_G(n)$	APTITUD
1	C_{10}	$\Theta(1)$	0
2	$C_{13}(1,3,4)$	$\Theta(n)$	0.918
3	$C_{13}(1,3,4) \boxtimes C_{13}(1,3,4)$	$\Theta(n^2)$	0.941
4	O_3	$\Theta \left(\sqrt{2}^{\sqrt{2}^{\dots^{\sqrt{2}^n}}} \right)$ 	0.270
5*	$C_{25}(1,3,6,7,8)$	$\Theta \left(3^{\frac{n}{2}} \right)$	0.997

Tabla 2: Gráficas con su función de crecimiento (cota ajustada asintótica) y la aptitud que se obtiene con la fórmula 3.8.

* Para la gráfica 5, en este trabajo se tiene la conjetura de que su crecimiento es exponencial: $\Theta \left(3^{\frac{n}{2}} \right)$.

$\frac{|Y_G|}{N+1}$, en la definición 3.8, sirve para penalizar a las gráficas cuya función $g_G(i)$, tiene crecimiento súper exponencial (con $n \leq N$): ya que en estos casos $|Y_G| < N+1$ y en consecuencia:

$$\frac{|Y_G|}{N+1} < \frac{N+1}{N+1} = 1,$$

por lo que el valor ρ_G^2 se disminuye al multiplicarse por $\frac{|Y_G|}{N+1}$, con lo que la aptitud es menor. Por otra parte, si $|Y_G| = N+1$, se tiene que:

$$\frac{|Y_G|}{N+1} = \frac{N+1}{N+1} = 1,$$

por lo que en este caso la aptitud de G está determinada completamente por ρ_G^2 .

En la tabla 2, se muestran algunas gráficas con su función de crecimiento y su aptitud, de acuerdo a la definición 3.8. La gráfica 1, es un ejemplo de gráfica con función de crecimiento constante. La gráfica 2 (gráfica circulante de 13 vértices y con saltos: 1,3 y 4), es un ejemplo de gráfica con crecimiento lineal [39]. La gráfica 3, que resulta del producto fuerte de dos gráficas con crecimiento lineal, es un ejemplo de gráfica con crecimiento polinomial (cuadrático). La gráfica 4, el octaedro, es un ejemplo de gráfica con crecimiento súper exponencial. Finalmente, la gráfica 5, es un ejemplo de gráfica que conjeturamos tiene crecimiento exponencial: note que de acuerdo a la definición 3.8, esta es la gráfica con mayor aptitud: note que la aptitud es casi 1, pero no 1, debido a la falta de precisión del modelo de regresión utilizado (esto se discutió al enunciar la fórmula 3.1).

En el algoritmo 2, se muestra el pseudocódigo de la implementación que se hizo en este trabajo, de la función de aptitud f_A , de acuerdo a la definición 3.8. La función *VectorDeCrecimiento*, que está en la línea 7, del algoritmo 2, es la función que se definió en el algoritmo 1.

El cálculo de la aptitud de una gráfica G , se hace de las líneas 5 a la 14, del algoritmo 2. Una vez que se ha calculado la aptitud, esta se almacena dentro de la gráfica G (línea 15): esto evita que se tenga que repetir el mismo cálculo si se invoca la función *Aptitud* con la misma gráfica G ; en cuyo caso sólo se ejecutan las líneas 2, 3 y 17.

Algoritmo 2: cálculo de la aptitud de una gráfica

Entrada: una gráfica G .

Salida: la aptitud de la función G , de acuerdo a la definición 3.8.

```

1  Función Aptitud( $G$ )
2      si  $G$ .aptitud existe entonces
3          | aptitud =  $G$ .aptitud
4      en otro caso
5          |  $N = 7$ 
6          |  $M = 4$ 
7          |  $Y_G = \text{VectorDeCrecimiento}(G, N, M)$ 
8          |  $X_G = [0, \dots, |Y_G| - 1]$ 
9          | si  $\text{Var}(X_G) \neq 0$  y  $\text{Var}(Y_G) \neq 0$  entonces
10             |  $\rho_G = \frac{\text{cov}(X_G, Y_G)}{\sqrt{\text{Var}(X_G) \cdot \text{Var}(Y_G)}}$ 
11             | en otro caso
12                 |  $\rho_G = 0$ 
13             | fin
14             | aptitud =  $\rho_G^2 \cdot \left(\frac{|Y_G|}{N+1}\right)$ 
15             |  $G$ .aptitud = aptitud
16      fin
17      regresar aptitud;
18 fin

```

En el algoritmo 3, se describe el pseudocódigo de la función *Aptitudes*, la cual permite calcular las aptitudes de una colección de gráficas. La función *Aptitud*, en la línea 6, es la función que se describe en el algoritmo 2. La función *Aptitudes* se usa con frecuencia en el algoritmo genético de este trabajo.

Algoritmo 3: cálculo de aptitudes para una colección de gráficas.

Entrada: una lista $\text{gráficas} = [G_1, G_2, \dots, G_N]$, donde cada G_i es una gráfica de n vértices.

Salida: una lista $\text{aptitudes} = [f_A(G_1), f_A(G_2), \dots, f_A(G_N)]$.

```

1 Función Aptitudes(gráficas)
2    $N = |\text{gráficas}|$ 
3    $\text{aptitudes} = []$ 
4   para cada  $i \in \{1, 2, \dots, N\}$  hacer
5      $G_i = \text{gráficas}[i]$ 
6      $\text{aptitudes}[i] = \text{Aptitud}(G_i)$ 
7   fin
8   regresar  $\text{aptitudes}$ ;
9 fin
```

3.3 PASOS GENERALES DE UN ALGORITMO GENÉTICO

Una vez que se ha definido la codificación, decodificación y función de aptitud, lo siguiente es implementar el algoritmo genético, siguiendo una heurística que trabaja sobre un conjunto de soluciones candidatas, que se denominan como *población*. El algoritmo genético genera y modifica dicha población mediante los siguientes pasos [24]:

1. *Inicialización*: crea una población inicial.
2. *Reproducción*: elige pares de soluciones de la población de acuerdo a su *aptitud*.
3. *Cruzamiento*: combina pares de soluciones para formar nuevas soluciones.
4. *Mutación*: modifica aleatoriamente las características de las nuevas soluciones.
5. *Depredación*: elimina soluciones con poca *aptitud* de la población.

El diagrama de flujo de la figura 3, muestra el orden en que se ejecutan los pasos generales de un algoritmo genético. Cada uno de estos pasos y las particularidades de su implementación, se explican a detalle en las siguientes secciones.

Observe que una vez que se realiza el paso de *Inicialización* (o el paso de *Depredación*), se revisa si las soluciones candidatas cumplen ciertos requisitos (figura 3, diamante verde): por ejemplo, se puede revisar si se ha encontrado una solución candidata con una aptitud que rebese algún umbral preestablecido. Si se cumplen los requisitos, el algoritmo reporta los

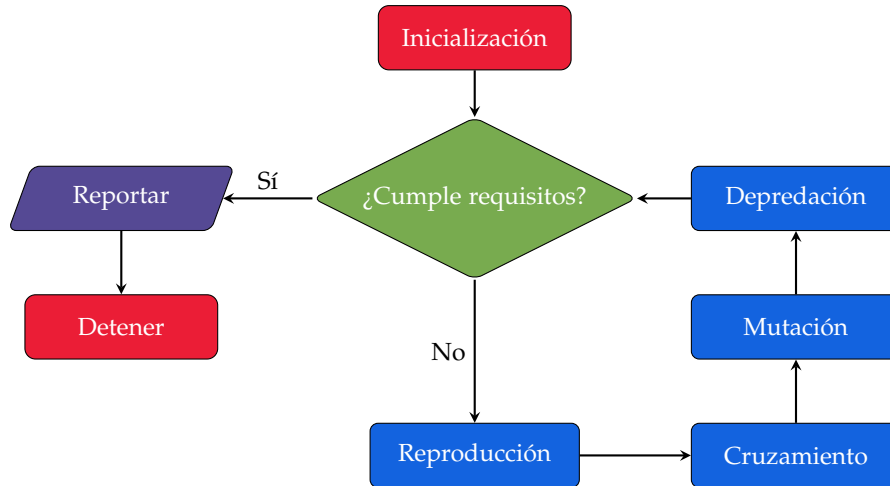


Figura 3: Diagrama de flujo de un algoritmo genético.

valores encontrados y termina (figura 3, recuadros morado y rojo); en caso contrario, se repiten de nuevo los pasos generales del algoritmo genético.

3.3.1 Inicialización

La *inicialización* es el primer paso que se realiza en un algoritmo genético y consiste en generar una población inicial de soluciones candidatas. Usualmente cada solución candidata que es parte de esta población inicial, se generan de manera aleatoria.

inicialización

En el caso de este trabajo, el proceso de generar una gráfica aleatoria se simplifica al usar la función RandomGraph de YAGS, la cual permite generar gráficas aleatorias de n vértices.

*función
RandomGraph de
YAGS*

3.3.2 Reproducción

En la selección natural, los organismos que tienen mayor aptitud tienen mayor probabilidad de reproducirse y tener descendencia [45]. De manera similar, en un algoritmo genético las soluciones candidatas que tienen mayor aptitud, también tienen mayor probabilidad de ser elegidas en el paso de *reproducción*. En los AGs, este paso consiste en seleccionar de manera aleatoria pares de soluciones candidatas de la población, las cuales generarán descendientes como resultado del *cruzamiento* y la *mutación* (pasos que se explican en las siguientes secciones).

*selección aleatoria de
una solución
candidata*

El proceso para elegir una solución candidata de manera aleatoria, se hace de la siguiente manera:

1. Se obtiene la suma T , de todas las aptitudes de la población. Por ejemplo, en la tabla 3, se muestra una población de 4 soluciones candidatas del espacio de búsqueda de gráficas de 4 vértices. Observe que la suma de las aptitudes de la población en la tabla 3, es $T = 2.21$. También se muestra el porcentaje que representa cada aptitud del total T .
2. Se hace una partición del intervalo $[0, T]$, en segmentos de longitudes iguales a la aptitud de cada una de las soluciones candidatas de la población. Por ejemplo, en la figura 4, se muestra el intervalo $[0, T]$, que corresponde a la población de la tabla 3. El intervalo se ha particionado en 4 segmentos: el segmento 1, que corresponde a la aptitud de la gráfica 1, tiene los valores $[0, 0.55]$ y es de longitud 0.55 (24.88 % de la longitud total T); el segmento 2, que corresponde a la aptitud de la gráfica 2, tiene los valores $(0.55, 1.13]$ y longitud 0.58; el segmento 3 (corresponde a la gráfica 3), tiene los valores $(1.13, 1.88]$ y el segmento 4 (corresponde a la gráfica 4), tiene los valores $(1.88, 2.21]$.
3. Se genera un número aleatorio en el rango $[0, T]$ y se elige a la gráfica de aquel segmento que contenga al número. Por ejemplo, si el intervalo es como el de la figura 4 y se genera el número aleatorio 0.98, este valor se encuentra contenido en el segmento 2, por lo que en este caso se elegiría a la gráfica 2.

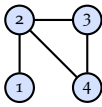
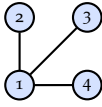
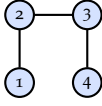
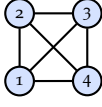
NUMERO	GRÁFICA	CODIFICACIÓN	APTITUD	PORCENTAJE
1		100111	0.55	24.88 %
2		111000	0.58	26.19 %
3		100101	0.75	33.97 %
4		111111	0.33	14.96 %
TOTAL			2.21	100 %

Tabla 3: Población de ejemplo en el espacio de búsqueda de gráficas de 4 vértices.

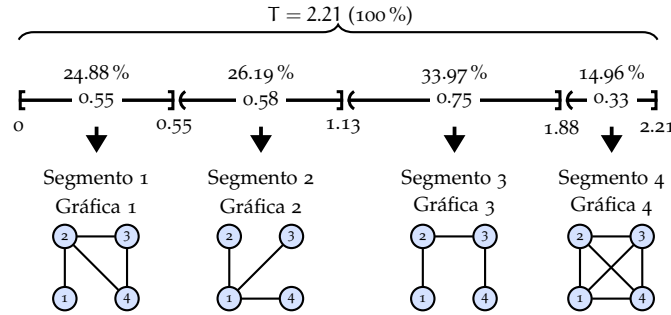


Figura 4: Segmento para elección aleatoria en un algoritmo genético.

Observe que la selección aleatoria de una solución candidata, se basa en particionar un intervalo $I = [0, T]$, en segmentos $S = \{s_1, s_2, \dots, s_n\}$ y elegir el índice $i \in \{1, 2, \dots, n\}$, del segmento s_i que contenga a un número aleatorio $r \in I$. El pseudocódigo de este proceso, se muestra en el algoritmo 4. Este algoritmo también se utiliza en el paso de depredación, que se explica más adelante, en la sección 3.3.5.

La función Sum en la línea 3, del algoritmo 4, es una función de GAP, que regresa el resultado de sumar todos los elementos de una lista (en este caso longSegmentos).

función Sum de GAP

Algoritmo 4: selección aleatoria por longitudes de segmentos.

Entrada: una lista longSegmentos = $[l_1, l_2, \dots, l_n]$, que representa las longitudes $l_i = |s_i|$, de un conjunto de segmentos $S = \{s_1, s_2, \dots, s_n\}$, que forman una partición de un intervalo $[0, T]$, con $T = \sum_{i=1}^n l_i$.

Salida: el índice i de aquel segmento s_i , que contiene a un número aleatorio generado en el intervalo $[0, T]$.

```

1  Función SelecciónAleatoria(longSegmentos)
2      R = 100000
3      T = Sum(longSegmentos)
4       $r = \frac{\text{Random}(0, R)}{R} \cdot T$ 
5      finSegmento = 0
6      i = 0
7      repetir
8          i = i + 1
9          finSegmento = finSegmento + longSegmentos[i]
10     hasta que finSegmento  $\geq r$ 
11     regresar i

```

Observación 3.9. La instrucción $\text{Random}(0, R)$, en la línea 4, del algoritmo 4,

función Random de GAP

es una función de GAP que genera un número aleatorio con una distribución uniforme, en el intervalo discreto $I = [0, 1, \dots, R]$. Por lo que el número

$$r = \text{Random}(0, R) \cdot \frac{T}{R},$$

cumple que:

$$r \in I \cdot \frac{T}{R} = \left[0 \cdot \frac{T}{R}, 1 \cdot \frac{T}{R}, \dots, R \cdot \frac{T}{R} \right] \subset [0, T].$$

Observe que las instrucciones en las líneas 5 a 10, del algoritmo 4, determinan la posición que tiene en la lista longSegmentos, aquel segmento $s_i \subseteq [0, T]$, que contiene al número r . Es decir, el algoritmo 4, regresa el índice i , de un segmento s_i seleccionado aleatoriamente.

Observación 3.10 (Probabilidad de que SeleccionAleatoria elija el índice i). Por la observación 3.9, el número aleatorio r que se genera en la instrucción 4, del algoritmo 4, es un número en el intervalo $[0, T]$. Observe que en las líneas 5 a 10, del algoritmo 4, se determina el índice i , del segmento $s_i \subseteq [0, T]$, que contiene al número r de la siguiente forma:

- Cuando $i = 1$, se revisa si $r \in s_1 = [0, l_1]$, si es el caso se regresa $i = 1$, en la línea 11, del algoritmo 4. Note que la longitud de s_1 es $|s_1| = l_1$.
- Si $i > 1$, se revisa si se cumple que:

$$r \in s_i = \left(\sum_{k=1}^{i-1} l_k, \sum_{k=1}^i l_k \right],$$

si es el caso se regresa i , en la línea 11, del algoritmo 4. Note que la longitud de s_i esta dada por: $|s_i| = \sum_{k=1}^i l_k - \sum_{k=1}^{i-1} l_k = l_i$.

Como r se genera con una distribución uniforme en el intervalo $[0, T]$ y por la granularidad de R , la probabilidad de que el algoritmo 4, regrese el índice i , es aproximadamente la probabilidad de que el segmento s_i contenga a r , es decir:

$$P_S(i) \approx \frac{|s_i|}{T} = \frac{l_i}{T}.$$

Para los fines de este trabajo esa aproximación fue suficiente para obtener resultados satisfactorios.

El pseudocódigo del algoritmo que se usa para elegir soluciones candidatas para el paso de reproducción, se muestra en el algoritmo 5: la función Aptitudes en la línea 2, es la función que se definió en el algoritmo 3; la función SelecciónAleatoria, en la línea 3, es la función que se definió en el algoritmo 4.

Algoritmo 5: selección aleatoria de una solución candidata para reproducción.

Entrada: una lista población, que contiene a las gráficas de la población actual.

Salida: la gráfica G , que se seleccionó aleatoriamente de la población para reproducción.

```

1 Función SeleccionarParaReproducción(población)
2   aptitudes = Aptitudes(población)
3    $i = \text{SelecciónAleatoria}(\text{aptitudes})$ 
4    $G = \text{población}[i]$ 
5   regresar  $G$ 

```

Observación 3.11 (Probabilidad de reproducción de una solución candidata). Sea la listas de gráficas:

$$\text{población} = [G_1, G_2, \dots, G_n],$$

que recibe la función *SeleccionarParaReproducción* del algoritmo 5.

Observe que en la línea 2, del algoritmo 5, se hace uso de la función *Aptitudes* (que se definió en el algoritmo 3), la cual construye la lista:

$$\text{aptitudes} = [f_A(G_1), f_A(G_2), \dots, f_A(G_N)].$$

En la línea 3, del algoritmo 5, la lista *aptitudes* se pasa a la función *SelecciónAleatoria* (que se definió en el algoritmo 4), la cual regresa un entero $i \in \{1, 2, \dots, N\}$, que se usa para elegir a la gráfica $G = G_i$. De esto se tiene que la probabilidad $P_R(G)$ de elegir a G , es igual a la probabilidad $P_S(i)$, de que la función *SelecciónAleatoria* elija el índice i . Por la Observación 3.10, se tiene que:

$$P_R(G) = P_S(i) \approx \frac{f_A(G)}{T},$$

donde $T = \sum_{k=1}^N f_A(G_k)$.

La función *SeleccionarParaReproducción* del algoritmo 5, se usa para elegir a una pareja de soluciones candidatas en el paso de reproducción. El pseudocódigo del paso de reproducción que se usó en este trabajo, se muestra en el algoritmo 6. La función *ShallowCopy*, que está en la línea 2, del algoritmo 6, es una función de GAP que crea una copia de la lista que recibe: en este caso se hace una copia de la lista población, que recibe como argumento la función *Reproducción*, del algoritmo 6. De esta forma las modificaciones que hace la función *Reproducción* a la lista población, no afectan a la población original del algoritmo genético.

*Función
ShallowCopy de
GAP*

Observe que el algoritmo 6, elige una gráfica G_i en la línea 3 y esa gráfica se remueve de la población en la línea 4: esto se hace para asegurar que la que la gráfica G_j que se elige en la línea 5, es diferente de la gráfica G_i .

Algoritmo 6: selección aleatoria de una pareja en el paso de reproducción.

Entrada: una lista población = $[G_1, G_2, \dots, G_n]$, que contiene a todas las gráficas G_i , que son parte de la población del algoritmo genético.

Salida: una lista pareja = $[G_i, G_j]$, que contiene a las gráficas G_i y G_j , con $G_i \neq G_j$, que fueron seleccionadas en el paso de reproducción.

```

1 Función Reproducción(población)
2   población = ShallowCopy(población)
3    $G_i = \text{SeleccionarParaReproducción}(\text{población})$ 
4   población = población  $\setminus G_i$ 
5    $G_j = \text{SeleccionarParaReproducción}(\text{población})$ 
6   pareja =  $[G_i, G_j]$ 
7   regresar pareja

```

3.3.3 Cruzamiento

cruzamiento En los AGs, el paso de *cruzamiento* consiste en generar una nueva solución candidata a partir de la pareja de soluciones candidatas, que fueron seleccionadas en el paso de reproducción. A estas nuevas soluciones se les conoce como *descendientes*. El paso de cruzamiento es análogo al proceso de *entrecruzamiento genético* que ocurre en los organismos vivos, en el cual se generan descendientes cuya información genética resulta de combinar segmentos de los genes de sus progenitores.

Sean $A = a_1 a_2 \dots a_n$ y $B = b_1 b_2 \dots b_n$, las cadenas binarias que codifican a una pareja de soluciones candidatas (a_i y $b_i \in \{0, 1\}$, para $1 \leq i \leq n$). El proceso de cruzamiento en los AGs elige aleatoriamente un índice $1 \leq k \leq n$ y a partir de las cadenas A y B , se forman dos nuevos descendientes con codificaciones: $H_1 = a_1 a_2 \dots a_k b_{k+1} \dots b_n$ y $H_2 = b_1 b_2 \dots b_k a_{k+1} \dots a_n$. Si bien este tipo de cruzamiento fue el que se usó en este trabajo y dio resultados satisfactorios (permitió encontrar la gráfica $C_{25}(1, 3, 6, 7, 8)$, que conjeturamos tiene crecimiento exponencial), experimentalmente encontramos que para el problema de búsqueda de esta investigación, este tipo de cruzamiento suele generar soluciones candidatas con baja aptitud, lo que puede hacer que el algoritmo genético nunca encuentre una solución candidata con una aptitud mínima deseada.

Para este trabajo, los experimentos computacionales parecen mostrar que un cruzamiento que da mejores resultados es el siguiente: dadas las cadenas binarias $A = a_1 a_2 \dots a_n$ y $B = b_1 b_2 \dots b_n$, se produce una nueva

solución candidata con codificación binaria $C = c_1 c_2 \dots c_n$, en la que cada dígito c_i , resulta de elegir aleatoriamente con una distribución uniforme, uno de los valores del conjunto: $\{a_i, b_i\}$. El pseudocódigo de este proceso se muestra en el algoritmo 7.

La función Random que está en la línea 6, del algoritmo 7, recibe como argumento una lista y elige aleatoriamente uno de los valores de la lista. Una variante de la función Random que recibe dos enteros en lugar de una lista, se describió en la observación 3.9.

Algoritmo 7: cruzamiento de dos cadenas binarias para producir un descendiente.

Entrada: Las listas $\text{cadenaA} = [a_1, a_2, \dots, a_n]$ y $\text{cadenaB} = [b_1, b_2, \dots, b_n]$, con $a_i, b_i \in \{0, 1\}$, que representan las codificaciones binarias de dos gráficas G_A y G_B , de n vértices.

Salida: Una lista $\text{cadenaC} = [c_1, c_2, \dots, c_n]$, con $c_i \in \{a_i, b_i\}$, que representa la codificación binaria de la gráfica que resulta de cruzar cadenaA y cadenaB .

```

1 Función Cruzamiento( $\text{cadenaA}$ ,  $\text{cadenaB}$ )
2    $n = |\text{cadenaA}|$ 
3    $\text{cadenaC} = []$ 
4   para cada  $i \in \{1, 2, \dots, n\}$  hacer
5      $\text{dígitos} = [\text{cadenaA}[i], \text{cadenaB}[i]]$ 
6      $\text{cadenaC}[i] = \text{Random}(\text{dígitos})$ 
7   regresar  $\text{cadenaC}$ 

```

3.3.4 Mutación

La *mutación* en los AGs es análoga a la mutación genética que ocurre en el proceso de evolución: las mutaciones son cambios que se producen al azar en la información de los genes de los descendientes que resultan del entrecruzamiento genético; estas mutaciones pueden ser mortales para el organismo descendiente, sin embargo, también pueden producir alelos que otorguen una mayor capacidad de adaptación. En condiciones normales, la probabilidad de que ocurra una mutación en un gen es muy baja.

mutación

De manera similar, en los AGs la mutación consiste en modificar aleatoriamente ciertas características de la solución candidata que resulta del cruzamiento. Si se está usando una codificación binaria, usualmente este paso se realiza de la siguiente forma: se toma la cadena que resulta del cruzamiento y por cada dígito de la cadena, de manera aleatoria, se cambia

por el valor contrario (si es 1 se cambia por 0 y viceversa). La probabilidad de que ocurra una mutación en cada dígito está dada por una probabilidad de mutación $0 \leq p_m \leq 1$. El pseudocódigo del paso de mutación que se usó en este trabajo, se muestra en el algoritmo 8. Para este trabajo se usó una probabilidad de mutación $p_m = 0.001$.

Algoritmo 8: mutación en un algoritmo genético.

Entrada: La probabilidad de mutación $0 \leq p_m \leq 1$ y una lista $\text{cadena} = [d_1, d_2, \dots, d_n]$, que representa la codificación binaria de una solución candidata.

Salida: Una lista $\text{cadenaMutada} = [d'_1, d'_2, \dots, d'_n]$, que resulta de las mutaciones de la lista cadena .

```

1 Función Mutación( $p_m$ ,  $\text{cadena}$ )
2    $n = |\text{cadena}|$ 
3    $\text{cadenaMutada} = []$ 
4   para cada  $i \in \{1, 2, \dots, n\}$  hacer
5      $\text{dígito} = \text{cadena}[i]$ 
6      $\text{mutar} = \text{SelecciónAleatoria}([p_m, 1 - p_m])$ 
7     si  $\text{mutar} = 1$  entonces
8       si  $\text{dígito} = 1$  entonces
9          $\text{dígito} = 0$ 
10      en otro caso
11         $\text{dígito} = 1$ 
12       $\text{cadenaMutada}[i] = \text{dígito}$ 
13  regresar  $\text{cadenaMutada}$ 

```

La función SelecciónAleatoria en la línea 6, del algoritmo 8, es la función que se definió en el algoritmo 4. Note que por la Observación 3.10, la probabilidad de que dicha función asigne el valor 1, a la variable mutar (también en la línea 6), es *aproximadamente* igual al valor del parámetro p_m , que recibe como argumento la función Depredación, del algoritmo 9.

3.3.5 Depredación

depredación La *depredación* en los AGs consiste en reemplazar soluciones candidatas de la población con los descendientes que resultan del cruzamiento y la mutación. Este paso es análogo al proceso de depredación que ocurre en la selección natural: en el cual, los organismos con menor aptitud tienen menos probabilidades de sobrevivir. De manera similar, en el paso de depredación de los AGs, las soluciones candidatas con una menor aptitud

tienen más probabilidades de ser seleccionadas para ser reemplazadas (es decir, tienen menos probabilidades de sobrevivir).

La depredación en un algoritmo genético se hace de la siguiente forma: se toma uno de los descendientes y se elige de manera aleatoria una solución candidata de la población; si la solución seleccionada tiene menor aptitud que el descendiente, se reemplaza a dicha solución con el descendiente, en caso contrario se descarta el descendiente y se preserva en la población a la solución seleccionada. La selección aleatoria se hace con probabilidad directamente proporcional a la *incompetencia* de una solución candidata (la incompetencia es el inverso de la aptitud).

incompetencia

El pseudocódigo del paso de depredación que se usó en este trabajo se muestra en el algoritmo 9. La función Aptitud, en la línea 6, del algoritmo 9, es la función que se definió en el algoritmo 2. La función SelecciónAleatoria, en la línea 7, del algoritmo 9, es la función que se definió en el algoritmo 4.

Algoritmo 9: depredación en un algoritmo genético.

Entrada: Una lista población = $[G_1, G_2, \dots, G_N]$, en el que cada gráfica G_i tiene n vértices. Una gráfica G de n vértices, que representa un descendiente.

Salida: La lista población después de la depredación.

```

1 Función Depredación(población, G)
2   N = |población|
3   incompetencias = [ ]
4   para cada  $i \in \{1, 2, \dots, N\}$  hacer
5      $G_i = \text{población}[i]$ 
6      $\text{incompetencias}[i] = \frac{1}{\text{Aptitud}(G_i)}$ 
7    $i = \text{SelecciónAleatoria}(\text{incompetencias})$ 
8    $H = \text{población}[i]$ 
9   si  $\text{Aptitud}(G) > \text{Aptitud}(H)$  entonces
10     $\text{población}[i] = G$ 
11  regresar población

```

Observe que las líneas de la 2 a la 6, del algoritmo 9, se usan para calcular las incompetencias de cada gráfica G_i . Las líneas 7 y 8, se usan para elegir a la gráfica H de la población. Por la Observación 3.10, la probabilidad de elegir a la gráfica H , es inversamente proporcional a la aptitud de la gráfica H (o dicho de otra forma, es directamente proporcional a la incompetencia de la gráfica H). Las líneas 9 y 10, del algoritmo 9, reemplazan a la gráfica H por la gráfica G , si la aptitud de G es mayor que H .

3.4 ALGORITMO GENÉTICO UTILIZADO

El pseudocódigo del algoritmo genético que se usó en este trabajo se muestra en el algoritmo 10: las funciones Aptitudes, Reproducción, Cruzamiento, Mutación y Depredación, que usa el algoritmo genético, se definieron respectivamente en los algoritmos: 3, 6, 7, 8 y 9.

InicializarPoblación

Algoritmo 10: algoritmo genético utilizado.

Entrada: Un entero $n \geq 1$, que determina que el algoritmo genético realice búsquedas en el conjunto de gráficas de n vértices.

Salida: Una lista de gráficas.

```

1 Función AlgoritmoGenético( $n$ )
2   tamPoblación = 300
3   aptitudMinima = 0.99
4   probMutación = 0.001
5   población = InicializarPoblación(tamPoblación,  $n$ )
6   aptitudes = Aptitudes(población)
7   repetir
8     pareja = Reproducción(población)
9     gráficaA = pareja[1]
10    gráficaB = pareja[2]
11    cadenaA = CodificarGráfica(gráficaA)
12    cadenaB = CodificarGráfica(gráficaB)
13    cadenaC = Cruzamiento(cadenaA, cadenaB)
14    cadenaC = Mutación(probMutación, cadenaC)
15    gráficaC = DecodificarAGráfica(cadenaC)
16    población = Depredación(población, gráficaC)
17    aptitudes = Aptitudes(población)
18  hasta que Maximum(aptitudes) > aptitudMinima
19
20  Print("Se encontró gráfica con aptitud mínima deseada")
21  regresar población
  
```

*función Maximum
de GAP*

La función Maximum, en la línea 18, del algoritmo 10, es una función de GAP que regresa el valor máximo de una lista dada (en este caso, la mayor aptitud de la lista aptitudes).

Para buscar gráficas con el algoritmo 10, se hacen múltiples invocaciones de la función AlgoritmoGenético(n), con n en el rango: $1 \leq n \leq 30$. Para la mayoría de estos valores, el algoritmo genético no puede encontrar una

gráfica cuya aptitud satisfaga la aptitud mínima deseada (mayor que 0.99). Sin embargo, para $n \geq 25$, el algoritmo por lo general encuentra a la gráfica $C_n(1, 3, 6, 7, 8)$.

Al invocar la función $\text{VectorCrecimiento}(G, N, M)$, del algoritmo 1, con los parámetros: $G = C_{25}(1, 3, 6, 7, 8)$, $N = 7$ y $M = 4$, se obtiene el siguiente vector de crecimiento (definición 3.4):

$$Y_G = (3.21888, 3.91202, 4.60517, 5.16479, 5.78383, 6.30992, 6.90776, 7.42357).$$

Por lo que de acuerdo a la definición 2, la aptitud $f_A(G)$ de la gráfica G , tiene el siguiente valor:

$$f_A(G) = 0.9978.$$

En la tabla 4 se muestran los valores de $|K^n(G)|$, $\ln(|K^n(G)|)$ y de la regresión lineal $y(n) = a + bn$, para $0 \leq n \leq 12$. Los valores de las constantes a y b , se calcularon para aproximar la relación de los vectores:

$$X = (0, 1, 2, \dots, 12) \text{ y } Y = (|G|, |K(G)|, |K^2(G)|, \dots, |K^{12}(G)|).$$

Por el lema 4, los valores que se obtuvieron para las constantes a y b , para los vectores X y Y , son: $a = 3.32637$ y $b = 0.596962$.

Como la aptitud de la gráfica G es casi 1, es de esperarse que los valores en las columnas $y(G)$ y $\ln(|K^n(G)|)$, de la tabla 4, sean muy similares: esto también se puede observar en la figura 5, que muestran las gráficas de $\ln(|K^n(G)|)$ y de la regresión $y(n)$.

Los datos experimentales antes mencionados, sugieren que la gráfica G , que se encontró con el algoritmo genético, tiene crecimiento exponencial. Resultados experimentales similares se obtienen para $C_n(1, 3, 7, 8)$, con $n \geq 25$. Esto nos ha llevado a plantear la conjetura que se plantea en la siguiente sección.

n	$ K^n(G) $	$\text{Ln}(K^n(G))$	$y(n) = a + bn$
0	25	3.21888	3.32637
1	50	3.91202	3.92334
2	100	4.60517	4.5203
3	175	5.16479	5.11726
4	325	5.78383	5.71422
5	550	6.30992	6.31118
6	1000	6.90776	6.90814
7	1675	7.42357	7.50511
8	3025	8.01467	8.10207
9	5050	8.52714	8.69903
10	9100	9.11603	9.29599
11	15175	9.6274	9.89295
12	27325	10.2156	10.4899

Tabla 4: Valores de $\text{Ln}(|K^n(G)|)$ y de la regresión $y(n) = a + bn$, con $a = 3.32637$ y $b = 0.596962$.

3.5 SOBRE LA GRÁFICA $C_n(1, 3, 6, 7, 8)$

Para la gráfica $G = C_N(1, 3, 6, 7, 8)$, con $N = 25$, en la tabla 4, se reportaron los valores de $|K^n(G)|$, para $0 \leq n \leq 12$. Tablas similares, se hicieron para $25 \leq N \leq 40$. Al examinar los datos obtenidos de los experimentos computacionales sobre el comportamiento de $|K^n(G)|$, se llegó a la siguiente conjetura:

Conjetura 3.12 (Conjetura del crecimiento de $C_N(1, 3, 6, 7, 8)$). *Sea la gráfica $G = C_N(1, 3, 6, 7, 8)$, con $N \geq 25$. Sea la función $g(n) = |K^n(G)|$, entonces $g(n)$ cumple la siguiente relación recursiva:*

$$g(n) = \begin{cases} N, & \text{para } n = 0, \\ 2N, & \text{para } n = 1, \\ 3 \cdot g(n-2) + g(0), & \text{para } n \geq 2. \end{cases}$$

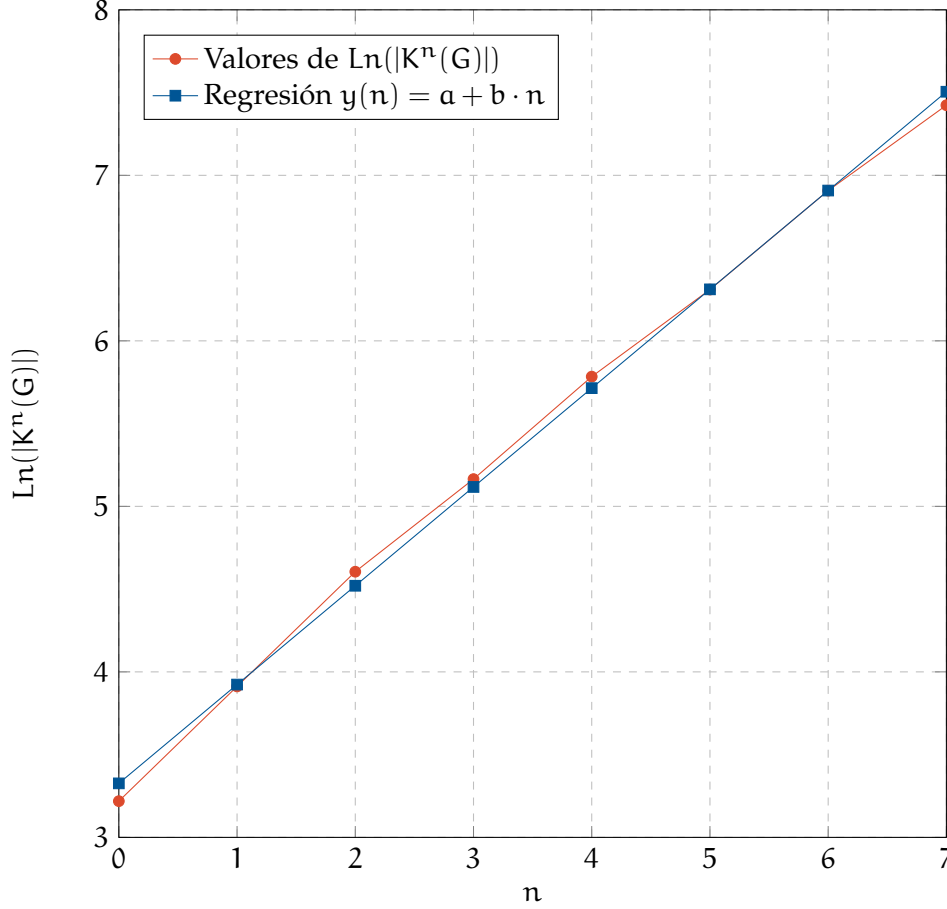


Figura 5: Gráfica de $\text{Ln}(|K^n(G)|)$ y de la regresión $y(n) = a + b \cdot n$.

Hemos podido verificar la conjetura 3.12, hasta $n \leq 12$. La relación de recurrencia $g(n)$ de la conjetura 3.12, está en función de $g(n-2)$. Sin embargo, se puede describir a $g(n)$ como una relación de recurrencia de $g(n-1)$, tal como se muestra en el siguiente lema.

Lema 3.13. Sea $g(n) = 3g(n-2) + g(0)$, con $g(1) = 2 \cdot g(0)$. Sea $f(n)$ la relación de recurrencia que se define como:

$$f(n) = \begin{cases} g(0), & \text{para } n = 0, \\ g(1), & \text{para } n = 1, \\ \frac{9 \cdot f(n-1) + 2 \cdot f(0)}{5}, & \text{para } n \text{ par, con } n \geq 2, \\ \frac{5 \cdot f(n-1) + f(0)}{3}, & \text{para } n \text{ impar, con } n \geq 3. \end{cases}$$

Entonces $g(n) = f(n)$, para $n \geq 0$.

Demostración. Por definición, $f(n) = g(n)$, para $0 \leq n \leq 1$. Para $n \geq 2$, la prueba se hará por inducción. Si $n = 2$, se tiene que:

$$g(2) = 3 \cdot g(2-2) + g(0) = 3 \cdot g(0) + g(0) = 4 \cdot g(0).$$

Además, como $f(1) = 2 \cdot g(0)$, se tiene que:

$$\begin{aligned} f(2) &= \frac{9 \cdot f(2-1) + 2 \cdot f(0)}{5} = \frac{9 \cdot f(1) + 2 \cdot f(0)}{5} = \frac{9(2 \cdot f(0)) + 2 \cdot f(0)}{5} \\ &= \frac{18 \cdot f(0) + 2 \cdot f(0)}{5} = \frac{20 \cdot f(0)}{5} = 4 \cdot f(0) = g(2). \end{aligned}$$

Para $n = 3$, se tiene que $g(3) = 3 \cdot g(3-2) + g(0) = 3 \cdot g(1) + g(0)$. Además, como $f(2) = \frac{9 \cdot f(1) + 2 \cdot f(0)}{5}$, se cumple que:

$$\begin{aligned} f(3) &= \frac{5 \cdot f(3-1) + f(0)}{3} = \frac{5 \cdot f(2) + f(0)}{3} = \frac{5 \left(\frac{9 \cdot f(1) + 2 \cdot f(0)}{5} \right) + f(0)}{3} \\ &= \frac{9 \cdot f(1) + 2 \cdot f(0) + f(0)}{3} = \frac{9 \cdot f(1) + 3 \cdot f(0)}{3} \\ &= 3 \cdot f(1) + f(0) = 3 \cdot g(1) + g(0) = g(3). \end{aligned}$$

Para $n \geq 4$, suponga que la afirmación del lema es cierta para $n-2$, es decir: $f(n-2) = g(n-2)$. Se tienen dos casos:

- *Caso 1: n es par.* Entonces $n-1$ es impar, luego:

$$f(n-1) = \frac{5 \cdot f((n-1)-1) + f(0)}{3} = \frac{5 \cdot f(n-2) + f(0)}{3}.$$

Por lo que al desarrollar $f(n)$ queda:

$$\begin{aligned} f(n) &= \frac{9 \cdot f(n-1) + 2 \cdot f(0)}{5} = \frac{9 \left(\frac{5 \cdot f(n-2) + f(0)}{3} \right) + 2 \cdot f(0)}{5} \\ &= \frac{3(5 \cdot f(n-2) + f(0)) + 2 \cdot f(0)}{5} = \frac{15 \cdot f(n-2) + 5 \cdot f(0)}{5} \\ &= 3 \cdot f(n-2) + f(0) = 3 \cdot g(n-2) + g(0) = g(n). \end{aligned}$$

- *Caso 2: n es impar.* Entonces $n-1$ es par, luego:

$$f(n-1) = \frac{9 \cdot f((n-1)-1) + 2 \cdot f(0)}{5} = \frac{9 \cdot f(n-2) + 2 \cdot f(0)}{5}.$$

Al sustituir en $f(n)$ queda:

$$\begin{aligned} f(n) &= \frac{5 \cdot f(n-1) + f(0)}{3} = \frac{5 \left(\frac{9 \cdot f(n-2) + 2 \cdot f(0)}{5} \right) + f(0)}{3} \\ &= \frac{9 \cdot f(n-2) + 2 \cdot f(0) + f(0)}{3} = \frac{9 \cdot f(n-2) + 3 \cdot f(0)}{3} \\ &= 3 \cdot f(n-2) + f(0) = 3 \cdot g(n-2) + g(0) = g(n). \end{aligned}$$

□

Las soluciones de la relación de recurrencia $g(n)$, se muestran en siguiente lema:

Lema 3.14. *Sea la relación de recurrencia $g(n) = 3g(n-2) + g(0)$, con $n \geq 0$ y $g(1) = 2 \cdot g(0)$. Entonces $g(n)$ tiene soluciones:*

$$g(n) = \begin{cases} \left(\frac{3^{\frac{n}{2}+1}-1}{2} \right) g(0), & \text{si } n \text{ es par,} \\ \left(\frac{5 \cdot 3^{\frac{n-1}{2}}-1}{2} \right) g(0), & \text{si } n \text{ es impar.} \end{cases}$$

Demostración. La prueba se hará por inducción sobre n . Para $n = 0$, como n es par, se tiene que:

$$g(0) = \left(\frac{3^{\frac{0}{2}+1}-1}{2} \right) g(0) = \left(\frac{3-1}{2} \right) g(0) = \frac{2}{2} \cdot g(0) = g(0).$$

Para $n = 1$, por definición $g(1) = 2 \cdot g(0)$ y como n es impar, se tiene que:

$$\begin{aligned} g(1) &= \left(\frac{5 \cdot 3^{\frac{1-1}{2}}-1}{2} \right) g(0) = \left(\frac{5 \cdot 3^0-1}{2} \right) g(0) = \left(\frac{5-1}{2} \right) g(0) \\ &= \frac{4}{2} \cdot g(0) = 2 \cdot g(0) = g(1) \end{aligned}$$

Para $n \geq 2$, suponga que la afirmación es cierta para $n-2$. Se tienen dos casos: n es par o n es impar.

Caso 1: n es par. Entonces $n-2$ es par y por la hipótesis de inducción:

$$g(n-2) = \frac{3^{\frac{n-2}{2}+1}-1}{2} = \frac{3^{\frac{n}{2}-1+1}-1}{2} = \frac{3^{\frac{n}{2}}-1}{2}.$$

Luego:

$$\begin{aligned} g(n) &= 3 \cdot g(n-2) + g(0) = 3 \cdot \left(\frac{3^{\frac{n}{2}}-1}{2} \right) \cdot g(0) + g(0) \\ &= \left(\frac{3 \cdot 3^{\frac{n}{2}}-3}{2} \right) g(0) + g(0) = \left(\frac{3^{\frac{n}{2}+1}-1-2}{2} \right) g(0) + g(0) \\ &= \left(\frac{3^{\frac{n}{2}+1}-1}{2} \right) g(0) - \frac{2}{2} \cdot g(0) + g(0) = \left(\frac{3^{\frac{n}{2}+1}-1}{2} \right) g(0) \end{aligned}$$

Caso 2: n es impar. Entonces $n - 2$ es impar y por la hipótesis de inducción:

$$\begin{aligned} g(n-2) &= \left(\frac{5 \cdot 3^{\frac{(n-2)-1}{2}} - 1}{2} \right) g(0) = \left(\frac{5 \cdot 3^{\frac{(n-1)-2}{2}} - 1}{2} \right) g(0) \\ &= \left(\frac{5 \cdot 3^{\frac{n-1}{2}-\frac{2}{2}} - 1}{2} \right) g(0) = \left(\frac{5 \cdot 3^{\frac{n-1}{2}-1} - 1}{2} \right) g(0). \end{aligned}$$

Luego:

$$\begin{aligned} g(n) &= 3 \cdot g(n-2) + g(0) = 3 \left(\frac{5 \cdot 3^{\frac{n-1}{2}-1} - 1}{2} \right) g(0) + g(0) \\ &= \left(\frac{3 \cdot 5 \cdot 3^{\frac{n-1}{2}-1} - 3}{2} \right) g(0) + g(0) = \left(\frac{5 \cdot 3^{\frac{n-1}{2}} - 3}{2} \right) g(0) + g(0) \\ &= \left(\frac{5 \cdot 3^{\frac{n-1}{2}} - 1 - 2}{2} \right) g(0) + g(0) = \left(\frac{5 \cdot 3^{\frac{n-1}{2}} - 1}{2} \right) g(0) - \frac{2}{2} \cdot g(0) + g(0) \\ &= \left(\frac{5 \cdot 3^{\frac{n-1}{2}} - 1}{2} \right) g(0) \end{aligned}$$

□

El siguiente teorema muestra que la cota ajustada asintótica de $g(n)$ es una función exponencial.

Teorema 3.15. Sea $g(n) = 3g(n-2) + g(0)$, con $g(0) > 0$ y $g(1) = 2 \cdot g(0)$. Entonces $g(n) = \Theta(3^{\frac{n}{2}})$.

Demostración. Por la definición de cota ajustada asintótica, para demostrar que $g(n) = \Theta(t^n)$, basta con mostrar que existen constantes positivas c_1 y c_2 y un valor n_0 , tal que:

$$c_1 \cdot 3^{\frac{n}{2}} < g(n) < c_2 \cdot 3^{\frac{n}{2}}, \text{ para } n \geq n_0$$

Se demostrará que: $n_0 = 0$, $c_1 = \frac{g(0)}{3}$ y $c_2 = 5 \cdot g(0)$. Para esto se tienen dos casos: n es par o n es impar.

Caso 1: n es par. Por el lema 3.14, para n par, con $n \geq 0$, se tiene que:

$$g(n) = \left(\frac{3^{\frac{n}{2}+1} - 1}{2} \right) g(0).$$

De la igualdad anterior, se obtiene la cota superior para $g(n)$:

$$\begin{aligned} g(n) &= \left(\frac{3^{\frac{n}{2}+1} - 1}{2} \right) g(0) < \left(\frac{3^{\frac{n}{2}+1}}{2} \right) g(0) = \frac{3^{\frac{n}{2}} \cdot 3 \cdot g(0)}{2} \\ &< \frac{3^{\frac{n}{2}} \cdot 10 \cdot g(0)}{2} = 5 \cdot g(0) \cdot 3^{\frac{n}{2}} = c_2 \cdot 3^{\frac{n}{2}} \end{aligned}$$

Para la cota inferior:

$$\begin{aligned} g(n) &= \left(\frac{3^{\frac{n}{2}+1} - 1}{2} \right) g(0) > \left(\frac{3^{\frac{n}{2}+1} - 3^{\frac{n}{2}}}{2} \right) g(0) = \left(\frac{3-1}{2} \right) 3^{\frac{n}{2}} \cdot g(0) \\ &= g(0) \cdot 3^{\frac{n}{2}} > \frac{g(0)}{3} \cdot 3^{\frac{n}{2}} = c_1 \cdot 3^{\frac{n}{2}} \end{aligned}$$

Caso 2: n es impar. Por el lema 3.14, para n impar, se tiene que:

$$g(n) = \left(\frac{5 \cdot 3^{\frac{n-1}{2}} - 1}{2} \right) g(0).$$

De la igualdad anterior, se obtiene la cota superior:

$$\begin{aligned} g(n) &= \left(\frac{5 \cdot 3^{\frac{n-1}{2}} - 1}{2} \right) g(0) < \left(\frac{5 \cdot 3^{\frac{n-1}{2}}}{2} \right) g(0) < 5 \cdot 3^{\frac{n-1}{2}} \cdot g(0) \\ &= 5 \cdot 3^{\frac{n}{2}} \cdot 3^{-\frac{1}{2}} \cdot g(0) = \left(\frac{5}{3^{\frac{1}{2}}} \right) \cdot g(0) \cdot 3^{\frac{n}{2}} < 5 \cdot g(0) \cdot 3^{\frac{n}{2}} = c_2 \cdot g(0) \end{aligned}$$

Para la cota inferior:

$$\begin{aligned} g(n) &= \left(\frac{5 \cdot 3^{\frac{n-1}{2}} - 1}{2} \right) g(0) > \left(\frac{5 \cdot 3^{\frac{n-1}{2}} - 3^{\frac{n-1}{2}}}{2} \right) g(0) \\ &= \left(\frac{5-1}{2} \right) \cdot 3^{\frac{n-1}{2}} \cdot g(0) = \frac{4}{2} \cdot g(0) \cdot 3^{\frac{n}{2}} \cdot 3^{-\frac{1}{2}} \\ &= \frac{2}{\sqrt{3}} \cdot g(0) \cdot 3^{\frac{n}{2}} > \frac{2}{3} \cdot g(0) \cdot 3^{\frac{n}{2}} > \frac{1}{3} \cdot g(0) \cdot 3^{\frac{n}{2}} = c_1 \cdot 3^{\frac{n}{2}}. \end{aligned}$$

De los dos casos anteriores se concluye que para $n \geq n_0$, con $n_0 = 0$, se tiene que:

$$c_1 \cdot 3^{\frac{n}{2}} \leq g(n) \leq c_2 \cdot 3^{\frac{n}{2}}.$$

Luego $g(n) = \Theta(3^{\frac{n}{2}})$. □

Si la conjetura 3.12 es verdadera, por el teorema 3.15, se tendría que la gráfica $G = C_N(1, 3, 6, 7, 8)$, con $N \geq 25$, cumpliría que:

$$|K^n(G)| = \Theta(a^n), \text{ con } a = \sqrt{3} > 1.$$

Es decir, G sería un ejemplo de una gráfica en la que sus gráficas iteradas de clanes tienen crecimiento exponencial.

Al estudiar las propiedades de las gráficas de clanes de $C_{25}(1, 3, 6, 7, 8)$, hemos encontrado una serie de resultados (que se presentan en el capítulo 4), que podrían ser de utilidad para probar la conjetura 3.12.

GRÁFICAS BOBINA GENERALIZADAS

En este capítulo se estudian las gráficas bobina generalizadas, las cuales son una generalización de gráficas circulantes (bajo ciertas restricciones, que se enuncian en el Lema 4.3) y en particular de $C_n(1, 3, 6, 7, 8)$ (sección 3.5). Los resultados que se presentan en este capítulo podrían ser útiles para demostrar la conjetura 3.4.

Se usará la notación habitual de grupos. En particular, Z_n denota al grupo de los enteros módulo n . Si $z \in Z_n$, se usará la notación $z = [z_0]$, con z_0 entero, para hacer énfasis en que z_0 es un representante de z .

También se usará la notación habitual para matrices, esto es, si T es una matriz, la entrada en el renglón i y columna j se denota por T_{ij} . Sin embargo, se usarán subconjuntos de Z_n como entradas de T (i.e. $T_{ij} \subseteq Z_n$). En particular $-T_{ij} = \{-z : z \in T_{ij}\}$ y se dice que T es antisimétrica, si $T_{ij} = -T_{ji}$, en tal caso $T_{ii} = -T_{ii}$.

4.1 DEFINICIÓN

A fin de enunciar la definición de gráfica bobina generalizada, primero es necesario definir la norma de Lee [14], para $[a] \in Z_n$, con $a \in \{0, 1, \dots, n-1\}$. Esta se define como:

$$\| [a] \| = \min\{n - a, a\}.$$

Es decir, si los elementos de Z_n son vértices de un ciclo C_n , la norma de Lee es la distancia más corta entre $[0]$ y $[a]$, en C_n . Observe que si $z \in Z_n \setminus \{0\}$, entonces $\|z\| > 0$.

Definición 4.1 (Gráfica bobina generalizada). *La gráfica G es una bobina generalizada, si sus vértices se pueden indexar de tal forma que se cumplen las siguientes condiciones (vea matriz de ejemplo 4.1 y Figura 6):*

*gráfica bobina
generalizada*

- (i). Z_n actúa libremente de tal forma que descompone a G en r orbitas $G^a = \{x_i^a : i \in Z_n\}$.

- (ii). Existe una matriz antisimétrica T (matriz de conexión), de tamaño $r \times r$, con entradas $T_{ij} \subseteq \mathbb{Z}_n$, tal que $x_i^a \sim x_j^b$ si y solo si $(j-i) \in T_{ab}$.
- (iii). $\max\{\|s\| : s \in T_{aa}\} < \frac{n}{3}$.

Observación 4.2. Si G es una gráfica bobina generalizada, se denotará al conjunto de sus r órbitas por $G/Z_n = \{G^1, G^2, \dots, G^r\}$.

La definición 4.1 es similar a la definición de gráfica bobina propuesta por Larrión, Pizaña y Villarroel-Flores, en [42]: con la diferencia de que una gráfica bobina debe cumplir restricciones adicionales. Es por esto que las gráficas que cumplen con la definición 4.1, las hemos denominado gráficas bobinas generalizadas.

Un ejemplo de gráfica bobina generalizada con 2 órbitas se obtiene al considerar la siguiente matriz de conexión T , cuyas entradas son subconjuntos de $\mathbb{Z}_n$, con $n \geq 22$:

$$T = \begin{pmatrix} \{\pm 1, \pm 6, \pm 7\} & \{-3, 1, 3, 4, 7\} \\ \{-7, -4, -3, -1, 3\} & \{\pm 3, \pm 6\} \end{pmatrix} \quad (4.1)$$

La gráfica bobina G que resulta de la matriz de conexión T , descrita en 4.1, se muestra en la Figura 6. Note que en dicha figura sólo se incluyen las adyacencias principales y el resto se obtiene al actuar $\mathbb{Z}_n$ sobre G . Observe que la órbita G^1 es isomorfa a $C_n(1, 6, 7)$ y la órbita G^2 es isomorfa a $C_n(3, 6)$. Note que las adyacencias de un vértice cualquiera en la órbita G^1 a la órbita G^2 , están dadas por la entrada T_{12} .

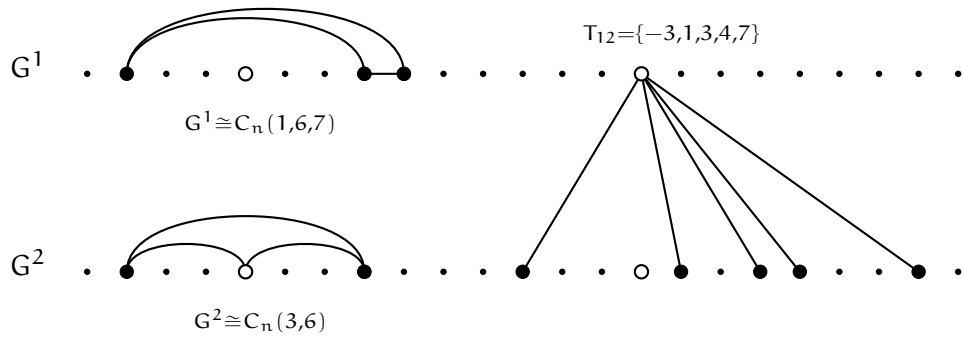


Figura 6: Ejemplo de gráfica bobina generalizada con 2 órbitas. Sólo se muestran las adyacencias esenciales, el resto se obtiene de que $\mathbb{Z}_n$ actúe sobre G . Los vértices O se usan como referencia.

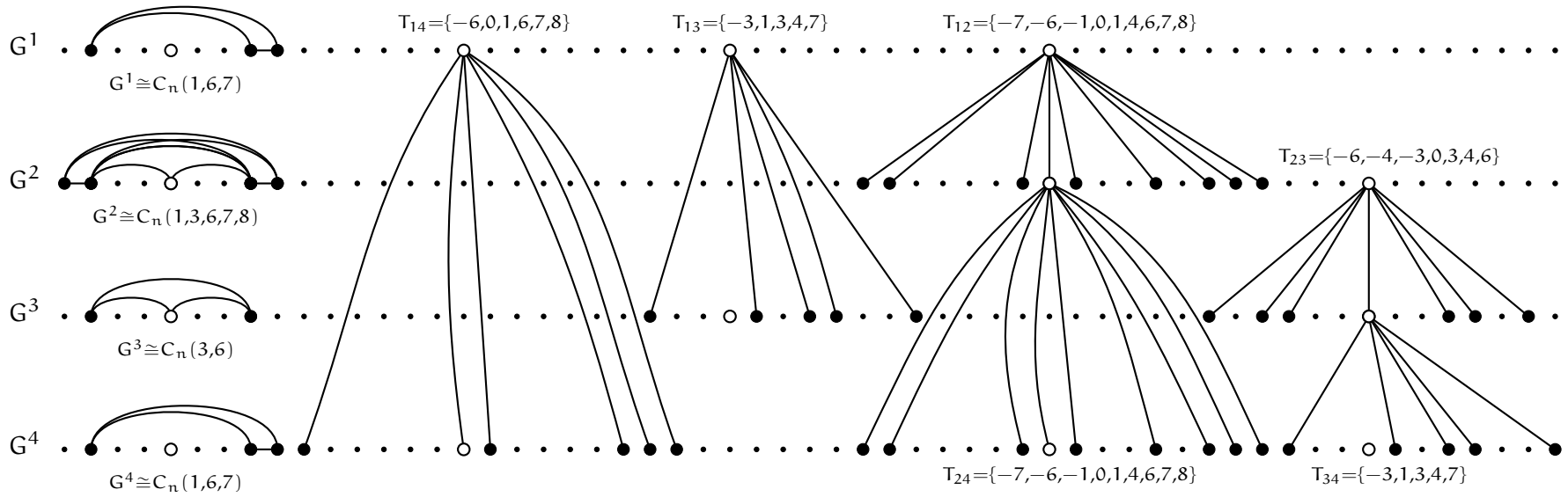


Figura 7: Ejemplo de gráfica bobina generalizada, con $G = K^2(C_n(1, 3, 6, 7, 8))$. Sólo se muestran las adyacencias esenciales, el resto se obtiene de que Z_n actúe sobre G . Los vértices O se usan como referencia.

Un ejemplo de gráfica bobina generalizada de 4 órbitas se muestra en la Figura 6, esta gráfica se obtiene al considerar $G = K^2(C_n(1, 3, 6, 7, 8))$, con $n \geq 25$ (por el Lema 4.3, que se enuncia más adelante, las gráficas iteradas de $C_n(1, 3, 6, 7, 8)$ son gráficas bobina generalizadas).

Puesto que $s \in T_{ab}$, determina que x_i^a y x_{i+s}^b , sean vértices adyacentes por una longitud $(i + s) - i = s$, en ocasiones denominaremos a los elementos de T_{ab} , como "saltos".

Lema 4.3. Sea $G = C_n(a_1, a_2, \dots, a_r)$ una gráfica circulante de n vértices y sea $J = \{\pm a_1, \pm a_2, \dots, \pm a_r\}$. Si $\max\{\|s\| : s \in J\} < \frac{n}{3}$, entonces G es una gráfica bobina generalizada.

Demostración. Se mostrará que G cumple las condiciones (i), (ii) y (iii), de la definición 4.1. Para la condición (i), por definición de gráfica circulante, Z_n actúa libremente en G y la descompone en órbitas $G/Z_n = \{G^1\}$, con $G^1 = G = \{x_0^1, x_1^1, \dots, x_{n-1}^1\}$.

Para la condición (ii), sea T , la matriz cuadrada 1×1 con entrada $T_{11} = J$. Como $J = -J$, entonces $T_{11} = -T_{11}$, luego T es antisimétrica. Además, por definición de gráfica circulante, $x_i^1 \sim x_j^1$ si y sólo si $j - i \in J = T_{11}$.

La condición (iii), $\max\{\|s\| : s \in J\} < \frac{n}{3}$, se cumple por hipótesis.

□

Por el lema 4.3, para $n \geq 25$, la gráfica $C_n(1, 3, 6, 7, 8)$ es una gráfica bobina generalizada. Es por ello que los resultados que se presentan en la siguiente sección, también se aplican para dicha gráfica.

4.2 RESULTADOS GENERALES

suma $S + j$ Si $S \subseteq Z_n$ y $j \in Z_n$, en esta sección se usará la notación $S + j$, para denotar al conjunto que resulta de sumar j a cada elemento de S , esto es: $S + j = \{i + j \in Z_n : i \in S\}$.

Puesto que la matriz de conexión cumple que $T_{aa} = -T_{aa}$, entonces $T_{aa} = \{\pm a_1, \pm a_2, \dots, \pm a_r\} \subseteq Z_n$. Note que cada órbita G^a es isomorfa a una gráfica circulante $C_n(a_1, a_2, \dots, a_r)$. Es por ello que el siguiente lema, es fundamental para estudiar a los clanes de G .

Lema 4.4. Sea C una completa de la gráfica circulante $C_n(a_1, a_2, \dots, a_r)$. Sea $J = \{\pm a_1, \pm a_2, \dots, \pm a_r\} \subseteq \mathbb{Z}_n$ y sea $z \in \mathbb{Z}_n \setminus \{0\}$, tal que $C = C + z$, entonces existe un $s \in J$, tal que $\|s\| \geq \frac{n}{3}$.

Demostración. Sea x_i un vértice de C . Por hipótesis $C = C + z$ y como $z \neq 0$, se tiene que x_{i+z} también es un vértice de C , con $x_i \neq x_{i+z}$. Por lo que $|C| \geq 2$.

Dependiendo del valor de $\|z\|$, se tienen dos casos: $\|z\| > \frac{n}{3}$ o $\|z\| \leq \frac{n}{3}$.

Caso 1: $\|z\| > \frac{n}{3}$.

Como C es una completa que contiene a los vértices x_i y x_{i+z} , se tiene que $x_i \sim x_{i+z}$ y por la definición de gráfica circulante: $(i + z) - i = z \in J$. Por lo que $s = z$, cumple que $s \in J$ y $\|s\| = \|z\| > \frac{n}{3}$.

Caso 2: $\|z\| \leq \frac{n}{3}$.

Como $z \in \mathbb{Z}_n \setminus \{0\}$, existe un entero z_0 , con $z = [z_0]$, tal que:

$$\|z\| = \|[z_0]\| = z_0 \leq \frac{n}{3} \quad (4.2)$$

Sea I_C , el conjunto de índices de los vértices de C , es decir:

$$I_C = \{i \in \mathbb{Z}_n : x_i \in C\}.$$

Por hipótesis $C = C + z$ (igualdad de conjuntos), por lo que el mapeo $\phi : I_C \rightarrow I_C$, dado por $\phi(i) = i + z$, es una biyección de I_C en I_C (es decir, ϕ es una permutación). Por la definición de ϕ , para cualquier entero $r \geq 0$ se cumple que:

$$\begin{aligned} \phi^r(i) &= (\phi \circ \phi \circ \dots \circ \phi)(i) \quad (r \text{ veces la composición de } \phi) \\ &= i + [r] \cdot z, \text{ con } [r] \in \mathbb{Z}_n. \end{aligned}$$

Sea $x \in I_C$, como $|C| = |I_C| \geq 2$ y $z \neq 0$, el principio del palomar implica que existe un entero t , con $2 \leq t \leq |I_C|$, tal que $\phi^t(x) = x$. Como también se cumple que $\phi^t(x) = x + [t] \cdot z$, entonces $x = x + [t] \cdot z$ y por lo tanto $[t] \cdot z = 0$. De esto se tiene la siguiente cadena de implicaciones:

$$\begin{aligned} [t] \cdot z &= [t] \cdot [z_0] = [t \cdot z_0] = 0 \implies t \cdot z_0 \equiv 0 \pmod{n} \\ &\implies \text{existe un entero } m \geq 1, \text{ tal que:} \\ &\quad t \cdot z_0 = mn. \end{aligned}$$

Como $t \cdot z_0 = mn$ y $t \geq 2$, existe un entero t_0 , con $1 \leq t_0 < t$, tal que:

$$t_0 \cdot z < n \quad (4.3)$$

$$(t_0 + 1)z \geq n \quad (4.4)$$

Por definición, el piso $\lfloor t_0/2 \rfloor$ es un entero. Como $t_0 \geq 1$, se tiene que $t_0 \neq \lfloor t_0/2 \rfloor$. Por lo que los números:

$$\alpha = \phi^{\lfloor t_0/2 \rfloor}(x) = x + \lfloor \lfloor t_0/2 \rfloor \rfloor \cdot z \quad \text{y} \quad \beta = \phi^{t_0}(x) = x + \lfloor t_0 \rfloor \cdot z,$$

cumplen que $\alpha \neq \beta$. Por la definición de ϕ , la completa C contiene a los vértices x_α y x_β , con $x_\alpha \neq x_\beta$. Por lo que $x_\alpha^a \sim x_\beta^a$ y por definición de gráfica circulante $\beta - \alpha \in S$.

Por lo que $s = \beta - \alpha$, cumple que $s \in J$. Se mostrará que $\|s\| \geq \frac{n}{3}$.

Para reescribir a s en términos de su representante, se toma $z = [z_0]$, con lo que se obtiene:

$$\begin{aligned} s &= \beta - \alpha = (x + \lfloor t_0 \rfloor \cdot z) - (x + \lfloor \lfloor t_0/2 \rfloor \rfloor \cdot z) \\ &= x + \lfloor t_0 \rfloor \cdot z - x - \lfloor \lfloor t_0/2 \rfloor \rfloor \cdot z = \lfloor t_0 \rfloor \cdot z - \lfloor \lfloor t_0/2 \rfloor \rfloor \cdot z \\ &= (\lfloor t_0 \rfloor - \lfloor \lfloor t_0/2 \rfloor \rfloor) \cdot z = \lfloor t_0 - \lfloor t_0/2 \rfloor \rfloor \cdot z = \lfloor t_0 - \lfloor t_0/2 \rfloor \rfloor \cdot [z_0] \\ &= \lfloor (t_0 - \lfloor t_0/2 \rfloor) \cdot z_0 \rfloor. \end{aligned}$$

Luego $s_0 = (t_0 - \lfloor t_0/2 \rfloor) \cdot z_0$, es un representante de s , esto es: $s = [s_0]$. Se mostrará que $s_0 < n$. Como $\lfloor t_0/2 \rfloor \geq \frac{t_0-1}{2}$, entonces $-\lfloor t_0/2 \rfloor \leq -\frac{t_0-1}{2}$, lo cual implica que:

$$s_0 = (t_0 - \lfloor t_0/2 \rfloor) z_0 \leq \left(t_0 - \frac{t_0-1}{2} \right) z_0 = \frac{t_0 \cdot z}{2} + \frac{1}{2} \cdot z_0.$$

Por las desigualdades 4.2 y 4.3, se tiene que $z_0 \leq \frac{n}{3}$ y $t_0 \cdot z < n$, al sustituir en la desigualdad anterior queda:

$$s_0 < \frac{t_0 \cdot z}{2} + \frac{1}{2} \cdot z_0 \leq \frac{n}{2} + \frac{1}{2} \left(\frac{n}{3} \right) = \frac{n}{2} + \frac{n}{6} = \frac{4n}{6} = \frac{2n}{3}. \quad (4.5)$$

Por lo que $s_0 < n$ y se puede aplicar la norma de Lee a $s = \lfloor [s_0] \rfloor$. Como $\|s\| = \min\{n - s_0, s_0\}$, se tienen dos casos: la norma es $\|s\| = s_0$ o es $\|s\| = n - s_0$.

Caso 2.1: la norma es $\|s\| = s_0$.

Como $\lfloor \frac{t_0}{2} \rfloor \leq \frac{t_0}{2}$, entonces $-\lfloor \frac{t_0}{2} \rfloor \geq -\frac{t_0}{2}$, con lo que se obtiene:

$$\begin{aligned} \|s\| &= s_0 = \left(t_0 - \left\lfloor \frac{t_0}{2} \right\rfloor \right) z_0 \geq \left(t_0 - \frac{t_0}{2} \right) z_0 = \left(\frac{t_0}{2} \right) z_0 \\ &= \left(\frac{t_0 + 1 - 1}{2} \right) z_0 = \left(\frac{t_0 + 1}{2} \right) z_0 - \frac{z_0}{2}. \end{aligned}$$

Por la desigualdad 4.2, se tiene que $z_0 \leq \frac{n}{3}$, por lo que $-z_0 \geq -\frac{n}{3}$. Además, por la desigualdad 4.4, $t_0 + 1 \geq n$, de esto se obtiene:

$$\|s\| \geq \left(\frac{t_0 + 1}{2}\right) z_0 - \frac{z_0}{2} \geq \frac{n}{2} - \frac{1}{2} \left(\frac{n}{3}\right) = \frac{n}{2} - \frac{n}{6} = \frac{2n}{6} = \frac{n}{3}.$$

Caso 2.2: la norma es $\|s\| = n - s_0$.

Por la desigualdad 4.5, se tiene que $s_0 < \frac{2n}{3}$, entonces $-s_0 > -\frac{2n}{3}$, por lo que:

$$\|s\| = n - s_0 > n - \frac{2n}{3} = \frac{n}{3}$$

De los casos 2.1 y 2.2, se concluye que $\|s\| \geq \frac{n}{3}$. $\square$

Para el resto de esta sección, G denota a una gráfica bobina generalizada. Se usará la notación de la definición 4.1, para denotar a los vértices, órbitas y matriz de conexión de G .

Algunas notaciones adicionales se muestran a continuación. Sea H una subgráfica de G y sea r el total de órbitas de G . El conjunto de los índices de los vértices de H , está dado por:

Índices $I(H)$

$$I(H) = \{i \in \mathbb{Z}_n : x_i^a \in H, \text{ para alguna } 1 \leq a \leq r\}.$$

En particular, note que $|I(H \cap G^a)| = |H \cap G^a|$. Por otra parte, sea $j \in \mathbb{Z}_n$, entonces $H + j$ denota el conjunto de vértices que resultan de trasladar j veces, a cada vértice de H , es decir:

vértices $H + j$

$$H + j = \{x_{i+j}^a : x_i^a \in H\}.$$

Finalmente, si P y Q son subconjuntos de $\mathbb{Z}_n$, se denotará con $P - Q$, al conjunto de todas las diferencias entre los elementos de $P \times Q$, esto es:

diferencias $P - Q$

$$P - Q = \{p - q \in \mathbb{Z}_n : p \in P \text{ y } q \in Q\}.$$

Observe que $-(P - Q) = Q - P$ y si $P \neq \emptyset$, entonces $P - P \neq \emptyset$.

Teorema 4.5. Sea q un clan de G , entonces para todo $k \in \mathbb{Z}_n \setminus \{0\}$, se tiene que $q + k$ es un clan de G , con $q \neq q + k$.

Demostración. Sean x_i^a y x_j^b vértices del clan q . Por definición $x_i^a \sim x_j^b$ si y solo si $j - i \in T_{ab}$. Como $j - i = j - i + k - k = (j + k) - (i + k)$, se tiene que $x_i^a \sim x_j^b$ si y solo si $x_{i+k}^a \sim x_{j+k}^b$. Luego $q + k$ es una completa y por la maximalidad de q , $q + k$ también es un clan.

Sea r el total de órbitas de G y sea $q^a = q \cap G^a$. Observe que la órbita G^a es isomorfa a una gráfica circulante con saltos $J = T_{aa}$. Entonces, si $q^a \neq \emptyset$,

por la contrapositiva del lema 4.4, la condición $\max\{\|s\| : s \in T_{aa}\} < \frac{n}{3}$, implica que $q^a \neq q^a + k$, para todo $k \in \mathbb{Z}_n \setminus \{0\}$. Como $q = \bigcup_{a=1}^r q_a$, se sigue que $q \neq q + k$, para todo $k \in \mathbb{Z}_n \setminus \{0\}$. $\square$

Lema 4.6. Sea q un clan de G , sean G^a y G^b orbitas de G , entonces:

$$I(G^b \cap q) - I(G^a \cap q) \subseteq T_{ab}.$$

Demostración.

Sea $s \in I(q \cap G^b) - I(q \cap G^a)$, entonces existen vértices $x_i^a, x_j^b \in q$, tales que $s = j - i$. Como q es un clan, se tiene que $x_i^a \sim x_j^b$, luego $j - i \in T_{ab}$. Por lo tanto $I(q \cap G^b) - I(q \cap G^a) \subseteq T_{ab}$. $\square$

Teorema 4.7. Sea G una gráfica bobina generalizada, entonces $K(G)$ también es una gráfica bobina generalizada.

Demostración. Se mostrarán que $K(G)$ cumple con las condiciones: (i), (ii) y (iii), de la definición 4.1, de gráfica bobina generalizada.

Para la condición (i).

Sea q_α un clan de G , el resultado de trasladar i veces, los vértices de q_α , esta dado por $X_i^\alpha = q_\alpha + i$. Por el teorema 4.5, el conjunto X_i^α es un clan de G y si $X_{i+z}^\alpha = X_i^\alpha$, para algún $z \in \mathbb{Z}_n$, entonces $z = 0$. De esto se tiene que $\mathbb{Z}_n$ actúa libremente en los clanes de G , por lo que existen clanes $R = \{q_1, q_2, \dots, q_t\}$, tales que $K(G)$ se descompone en t órbitas:

$$K(G)/\mathbb{Z}_n = \{K(G)^1, K(G)^2, \dots, K(G)^t\},$$

con $K(G)^\alpha = \{X_i^\alpha : i \in \mathbb{Z}_n\}$.

Para la condición (ii).

Para cada órbita $K(G)^\alpha$, tome el representante $q_\alpha = X_0^\alpha$. Los vértices de cada clan q_α , que están en la órbita G_a , están dados por $q_\alpha^a = G^a \cap q_\alpha$. Los índices de q_α^a , están dados por $I_\alpha^a = I(q_\alpha^a)$. Sea t el total de órbitas de $K(G)$ y sea $\mathcal{T}$, la matriz de tamaño $t \times t$ cuyas entradas se definen como:

$$\mathcal{T}_{\alpha\beta} = \bigcup_{k=1}^{|\mathbb{Z}_n|} (I_\alpha^k - I_\beta^k).$$

La matriz $\mathcal{T}$ cumple que:

$$\begin{aligned}\mathcal{T}_{\alpha\beta} &= \bigcup_{k=1}^{|G/Z_n|} (I_{\alpha}^k - I_{\beta}^k) = \bigcup_{k=1}^{|G/Z_n|} -(I_{\beta}^k - I_{\alpha}^k) \\ &= - \bigcup_{l=1}^{|G/Z_n|} (I_{\beta}^l - I_{\alpha}^l) = -\mathcal{T}_{\beta\alpha}.\end{aligned}$$

Por lo que $\mathcal{T}$ es una matriz antisimétrica. Se mostrará que $\mathcal{T}$ es la matriz de conexión de $K(G)$.

Afirmación 1: $X_i^{\alpha} \sim X_j^{\beta} \implies j - i \in \mathcal{T}_{\alpha\beta}$.

Como $X_i^{\alpha} = q_{\alpha} + i$ y $X_j^{\beta} = q_{\beta} + j$, la condición $X_i^{\alpha} \sim X_j^{\beta}$ implica que $X_i^{\alpha} \cap X_j^{\beta} = (q_{\alpha} + i) \cap (q_{\beta} + j) \neq \emptyset$. Por lo que existen vértices $x_a^c \in q_{\alpha}$ y $x_b^d \in q_{\beta}$, tales que $x_{a+i}^c = x_{b+j}^d$. Luego $c = d$ y se cumple que:

$$a + i = b + j \implies j - i = a - b.$$

Como $a \in I_{\alpha}^c = I(G^c \cap q_{\alpha})$ y $b \in I_{\beta}^c = I(G^c \cap q_{\beta})$, se sigue que:

$$j - i = a - b \in I_{\alpha}^c - I_{\beta}^c \subseteq \bigcup_{k=1}^{|G/Z_n|} (I_{\alpha}^k - I_{\beta}^k) = \mathcal{T}_{\alpha\beta}.$$

Afirmación 2: $j - i \in \mathcal{T}_{\alpha\beta} \implies X_i^{\alpha} \sim X_j^{\beta}$.

Como $j - i \in \mathcal{T}_{\alpha\beta}$, existe un entero $1 \leq c \leq |G/Z_n|$, tal que:

$$j - i \in I_{\alpha}^c - I_{\beta}^c = I(G^c \cap q_{\alpha}) - I(G^c \cap q_{\beta}).$$

De esto se tiene que existen vértices $x_a^c \in q_{\alpha}$ y $x_b^c \in q_{\beta}$, tales que $a - b = j - i$, por lo que $a + i = b + j$ y en consecuencia $x_{a+i}^c = x_{b+j}^c$. Como $x_{a+i}^c \in q_{\alpha} + i = X_i^{\alpha}$ y $x_{b+j}^c \in q_{\beta} + j = X_j^{\beta}$, se tiene que $X_i^{\alpha} \cap X_j^{\beta} \neq \emptyset$, luego $X_i^{\alpha} \sim X_j^{\beta}$.

Por la afirmaciones 1 y 2, se tiene que $X_i^{\alpha} \sim X_j^{\beta}$ si y sólo si $j - i \in \mathcal{T}_{\alpha\beta}$.

Para la condición (iii).

Sea G^k cualquier órbita de G y sea α un entero con $1 \leq \alpha \leq |K(G)/Z_n|$, por el lema 4.6, se cumple lo siguiente:

$$I_{\alpha}^k - I_{\alpha}^k = I(G^k \cap q_{\alpha}) - I(G^k \cap q_{\alpha}) \subseteq T_{kk}$$

Luego:

$$\mathcal{T}_{\alpha\alpha} = \bigcup_{k=1}^{|G/Z_n|} (I_{\alpha}^k - I_{\alpha}^k) \subseteq \bigcup_{k=1}^{|G/Z_n|} T_{kk}.$$

Por hipótesis $\max\{\|s\|: s \in T_{\alpha\alpha}\} < \frac{n}{3}$, para $1 \leq \alpha \leq |G/Z_n|$ y por la contención anterior se obtiene que:

$$\begin{aligned} \max\{\|s\|: s \in \mathcal{T}_{\alpha\alpha}\} &\leq \max\{\|s\|: s \in \bigcup_{k=1}^{|G/Z_n|} T_{kk}\} \\ &= \max_{1 \leq \alpha \leq |G/Z_n|} \{\|s\|: s \in T_{\alpha\alpha}\} < \frac{n}{3}. \end{aligned}$$

Por los incisos (i), (ii) y (iii), se concluye que $K(G)$ es una gráfica bobina generalizada. $\square$

Observación 4.8. Observe que en el teorema 4.7, en la condición (i), también se muestra que cada clan $X_0^{\alpha} \in K(G)$, es un representante de la órbita $K(G)^{\alpha}$ de $K(G)$.

Observación 4.9. Sea $I_{\alpha}^k = I(G^k \cap X_0^{\alpha})$. Note que de acuerdo a la condición (ii), del teorema 4.7, la matriz de conexión $\mathcal{T}$, de $K(G)$, es de la forma:

$$\mathcal{T}_{\alpha\beta} = \bigcup_{k=1}^{|G/Z_n|} (I_{\alpha}^k - I_{\beta}^k),$$

Observación 4.10. Note que en el teorema 4.7, al probar la condición (iii), también se demostró que para las matrices de conexión de T y $\mathcal{T}$, de G y $K(G)$, respectivamente, se cumple lo siguiente:

$$\mathcal{T}_{\alpha\alpha} \subseteq \bigcup_{\alpha=1}^{|G/Z_n|} T_{\alpha\alpha}.$$

Por el teorema 4.7, todas las gráficas iteradas de clanes $K^i(G)$ de G , también son gráficas bobina generalizadas. La notación que se usará con $K^i(G)$, es similar a la que se ha usado con G . Esto es, si r es el total de órbitas de $K^i(G)$, entonces:

$$K^i(G)/Z_n = \{(K^i(G))^1, (K^i(G))^2, \dots, (K^i(G))^r\},$$

son las órbitas de $K^i(G)$ y X_j^{α} , denota el vértice j en la órbita α , es decir $X_j^{\alpha} \in (K^i(G))^{\alpha}$. La matriz de conexión de $K^i(G)$ se denotará por $T(K^i(G))$.

Puesto que toda gráfica circulante C , que cumple con las condiciones del lema 4.3, es una gráfica bobina generalizada con matriz de conexión T , una pregunta que surge al examinar las gráficas iteradas de C , es si puede ocurrir que para alguna gráfica $K^i(C)$, alguna de sus órbitas $(K^i(C))^\alpha$, tiene un salto que no está en C : esto es, existe un $s \in T(K^i(G))_{\alpha\alpha}$, tal que $s \notin T(C)_{11}$. El siguiente lema muestra que dicha situación no ocurre.

Lema 4.11. *Sea $C = C_n(a_1, a_2, \dots, a_r)$ una gráfica circulante de n vértices, tal que para $J = \{\pm a_1, \pm a_2, \dots, \pm a_r\}$, se tiene que $\max\{\|s\| : s \in J\} \leq \frac{n}{3}$. Para $i \geq 0$ y para cualquier $1 \leq \alpha \leq |K^i(C)/Z_n|$, se cumple lo siguiente:*

$$T(K^i(C))_{\alpha\alpha} \subseteq J$$

Demostración. La demostración se hará por inducción sobre i . Si $i = 0$, entonces $K^0(C) = C$ y por el lema 4.3, la matriz de conexión de C es de tamaño 1×1 , con $T(C)_{11} = J$. Suponga que la afirmación es verdadera para $i \geq 0$. Por la observación 4.10, para cualquier $1 \leq \alpha \leq |K^{i+1}(C)/Z_n|$, se tiene que:

$$T(K^{i+1}(C))_{\alpha\alpha} \subseteq \bigcup_{a=1}^{|K^i(C)/Z_n|} T(K^i(C))_{a\alpha}.$$

Por la hipótesis de inducción, $T(K^i(G))_{a\alpha} \subseteq J$, para $1 \leq a \leq |K^i(G)/Z_n|$, al sustituir en la desigualdad anterior queda:

$$T(K^{i+1}(C))_{\alpha\alpha} \subseteq \bigcup_{a=1}^{|K^i(C)/Z_n|} J = J.$$

□

El siguiente lema muestra que los saltos de la matriz de conexión, se forman de las diferencias de los índices, de todos los clanes representantes $X_0^\alpha \in K(G)$.

Lema 4.12. *Sea $X_0^\alpha \in K(G)^\alpha$, sea $I_\alpha^a = I(G^a \cap X_0^\alpha)$ y $I_\alpha^b = I(G^b \cap X_0^\alpha)$. Sea T la matriz de conexión de G , entonces*

$$T_{ab} = \bigcup_{\alpha=1}^{|K(G)/Z_n|} (I_\alpha^b - I_\alpha^a).$$

Demostración. Por el lema 4.6, para todo $1 \leq \alpha \leq |K(G)/Z_n|$, se tiene que $I_\alpha^b - I_\alpha^a \subseteq T_{ab}$, luego:

$$\bigcup_{\alpha=1}^{|K(G)/Z_n|} (I_\alpha^b - I_\alpha^a) \subseteq \bigcup_{\alpha=1}^{|K(G)/Z_n|} T_{ab} = T_{ab}.$$

Para la otra contención, sea $s \in T_{ab}$ y sean los vértices $x_0^a \in G^a$ y $x_s^b \in G^b$. Como $s - 0 = s \in T_{ab}$, se tiene que $x_0^a \sim x_s^b$. Sea q un clan de G , tal que $q \supseteq \{x_0^a, x_s^b\}$. Como $K(G)$ es una gráfica bobina generalizada, existe un $j \in Z_n$ y un entero $1 \leq \beta \leq |K(G)/Z_n|$, tal que:

$$X_j^\beta \in (K(G))^\beta \quad y \quad X_j^\beta = q.$$

Como $X_0^\beta = X_j^\beta - j$, los vértices $x_{0-j}^a = x_{-j}^a$ y x_{s-j}^b , son vértices de X_0^β . Por lo que $x_{-j}^a \in G^a \cap X_0^\beta$ y $x_{s-j}^b \in G^b \cap X_0^\beta$, luego:

$$(s - j) - (-j) \in I(G^b \cap X_0^\beta) - I(G^a \cap X_0^\beta) = I_\beta^b - I_\beta^a.$$

Como $(s - j) - (-j) = s - j + j = s$, se tiene que:

$$s \in I_\beta^b - I_\beta^a \subseteq \bigcup_{\alpha=1}^{|K(G)/Z_n|} (I_\alpha^b - I_\alpha^a)$$

□

El siguiente lema relaciona adyacencias en órbitas de $K(G)$, con adyacentes en órbitas de G .

Lema 4.13. *Si $X_i^\alpha \sim X_j^\alpha$ en $K(G)$, existe una órbita G^a de G , tal que $x_i^a \sim x_j^a$ y $x_{-i}^a \sim x_{-j}^a$ en G .*

Demostración. Sean T y $\mathcal{T}$, las matrices de conexión de G y $K(G)$, respectivamente. Como $X_i^\alpha \sim X_j^\alpha$, entonces $j - i \in \mathcal{T}_{\alpha\alpha}$. Por la observación 4.10, se tiene que:

$$j - i \in \mathcal{T}_{\alpha\alpha} \subseteq \bigcup_{a=1}^{|G/Z_n|} T_{aa}.$$

Por lo que existe $1 \leq a \leq |G/Z_n|$, tal que $j - i \in T_{aa}$, lo que implica que los vértices $x_i^a, x_j^a \in G$, cumplen que $x_i^a \sim x_j^a$. Como $j - i = -i - (-j)$, también se cumple que $x_{-i}^a \sim x_{-j}^a$. □

El siguiente teorema permite obtener completas en $K(G)$, a partir de completas de G . Intuitivamente dicho teorema se puede visualizar de la siguiente forma (véase figura: 8): si se alinean los clanes representantes $q_\alpha = X_0^\alpha \in K(G)$, de la gráfica G y si se dicha alineación se eligen las completas en una órbita (por ejemplo G^1 , que se muestra en color azul en la figura 8), entonces en la gráfica $K(G)$, una completa se forma de "transponer" e invertir las completas de dicha órbita (en la figura 8), la columna en azul).

Teorema 4.14. Sea $1 \leq \alpha \leq |G/Z_n|$ y sea $I_\alpha^a = I(G^a \cap X_0^\alpha)$, con $X_0^\alpha \in K(G)$, para $1 \leq \alpha \leq |K(G)/Z_n|$. Sea el conjunto de vértices de $K(G)$, dado por:

$$Q_\alpha = \bigcup_{\alpha=1}^{|K(G)/Z_n|} \{X_i^\alpha \in K(G) : i \in -I_\alpha^a\},$$

entonces Q_α es una completa de $K(G)$.

Demostración. Se mostrará que cualquier par de vértices de Q_α , son adyacentes. Sean X_i^α y X_j^β vértices de Q_α , por definición se tiene que $j \in -I_\beta^a$ y $i \in -I_\alpha^a$, que su vez implica que $-i \in -(-I_\alpha^a) = I_\alpha^a$. Por lo cual:

$$j - i \in (-I_\beta^a - (-I_\alpha^a)) = I_\alpha^a - I_\beta^a.$$

Por la observación 4.9, se tiene que $I_\alpha^a - I_\beta^a \subseteq \mathcal{T}_{\alpha\beta}$, luego $j - i \in \mathcal{T}_{\alpha\beta}$ y por lo tanto $X_i^\alpha \sim X_j^\beta$. $\square$

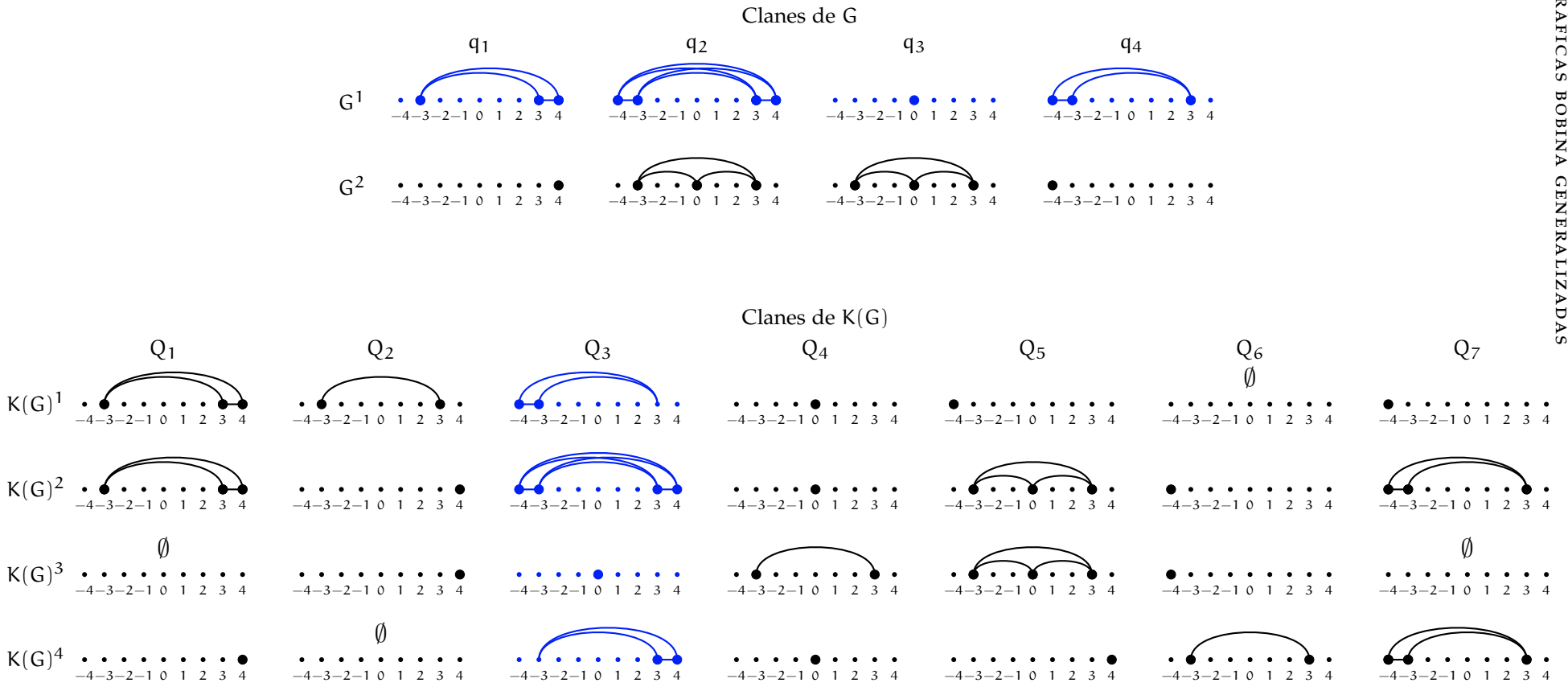


Figura 8: Clanes representantes de G y $K(G)$, para $G = K(C_n(1, 3, 6, 7, 8))$.

CONCLUSIONES

Este trabajo presenta resultados que aportan al estudio del problema de caracterizar las gráficas G , que maximizan a $|K^2(G)|$. Durante el proceso de investigación, se hicieron múltiples experimentos computacionales que llevaron a formular la conjetura 2.36, que afirma que $|K^2(G)|$ se maximiza cuando la gráfica $G = \mathcal{G}_n$ (definición 2.3). Puesto que la gráfica $\mathcal{G}_n$ es de la forma $\mathcal{G}_n = I_2 + H = S(H)$ (es una suspensión), esto nos llevó al estudio de las propiedades de la segunda gráfica iterada de clanes de suspensiones de gráficas, esto es: $K^2(S(G))$.

Para analizar a las gráficas $K^2(S(G))$, resultó muy útil usar un concepto nuevo de biclan y con ello el de gráfica de biclanes $B(G)$, con el cual se obtuvo la caracterización $B(K(G)) = K^2(S(G))$. Al estudiar las propiedades de la gráfica de biclanes $B(G)$, se llegó a una caracterización de las gráficas G que maximizan a $|B(G)|$. Se encontró que la gráfica $G = \mathcal{G}_n$, es la gráfica que maximiza a $|B(G)|$, lo cual es una evidencia más de que la conjetura 2.36, puede ser cierta.

De los resultados antes mencionados y de los que se mostraron en la sección 2.3, creemos que para caracterizar a las gráficas G que maximizan a $|K^2(G)|$, es esencial entender cómo cambia el número de clanes $|K^2(G)|$, con respecto a la gráfica G' que resulta de cambiar las adyacencias de dos de sus vértices $x, y \in G$, de tal forma que $G' = I_2 + G \setminus \{x, y\}$.

Por otra parte, el uso de experimentos computacionales mostró ser de extrema ayuda para formular o refutar afirmaciones, que fueron surgiendo en el transcurso de este trabajo. Creemos que el uso de paquetes computacionales como es el caso de GAP [23] y YAGS[8], facilita el explorar y entender la estructura combinatoria de diversos problemas en teoría de gráficas. En particular, en este trabajo se usaron los paquetes computacionales antes mencionados para estudiar gráficas de clanes.

Otro problema que se estudió en este trabajo, fue el buscar una gráfica G , tal que $|K^n(G)| = \Theta(a^n)$, con $a > 1$. La estrategia usada como punto de partida en este problema, fue el usar experimentos computacionales para buscar una gráfica con propiedades que sean de interés para la inves-

tigación. Esta es una estrategia general que creemos puede ser útil para estudiar otros problemas de gráficas.

Debido a que no es computacionalmente factible revisar a cada gráfica del espacio de búsqueda de gráficas de n vértices, fue necesario usar un tipo de heurística que permitiera a los experimentos computacionales hacer una búsqueda más dirigida, para encontrar una gráfica de interés. Para los fines de este trabajo, una heurística basada en algoritmos genéticos dio resultados satisfactorios para encontrar una gráfica G , que en los experimentos cumple que $|K^n(G)|$ tiene crecimiento exponencial. Dada una función $f(n)$, creemos que esta estrategia podría ser útil también para buscar gráficas cuyo orden de crecimiento de $|K^n(G)|$, sea $|K^n(G)| = \Theta(f(n))$.

Mediante una heurística basada en algoritmos genéticos, se encontró que experimentalmente la gráfica $G = C_n(1, 3, 6, 7, 8)$, con $n \geq 25$, cumple que $|K^n(G)| = \Theta(3^{n/2})$, lo que llevó a plantear la conjetura 3.12. A fin de estudiar esta conjetura, se propuso la definición de gráfica bobina generalizada, que resultó de analizar las propiedades de las gráficas iteradas de clanes de $C_n(1, 3, 6, 7, 8)$. Los resultados que se muestran en el capítulo 4, creemos que podrían ser de utilidad para probar la conjetura 3.12. Sin embargo, aún hace falta caracterizar la forma de los clanes de clanes de $C_n(1, 3, 6, 7, 8)$.

BIBLIOGRAFÍA

- [1] L. Alcón y M. Gutierrez. «A new characterization of clique graphs». En: *Matemática Contemporânea* 25 (ene. de 2003).
- [2] H.-J. Bandelt y E. Prisner. «Clique graphs and Helly graphs». En: *Journal of Combinatorial Theory, Series B* 51.1 (1991), págs. 34 -45. URL: <http://www.sciencedirect.com/science/article/pii/0095895691900044>.
- [3] N. A. Barricelli. «Esempi numerici di processi di evoluzione». En: *Methodos* (1954), págs. 45-68.
- [4] B. Beavis e I. Dobbs. En: *Optimisation and stability theory for Economic Analysis*. Cambridge University Press, 1990, 32–32.
- [5] B. Bollobás. *Combinatorics : set systems, hypergraphs, families of vectors, and combinatorial probability*. Cambridge Cambridgeshire New York: Cambridge University Press, 1986, pág. 82.
- [6] C. Bron y J. Kerbosch. «Finding all cliques of an undirected graph—Algorithm 457». En: *Communications of the ACM* 16 (1971), págs. 575-577.
- [7] B. Buchberger, G. E. Collins, R. Loos y R. Albrecht. «Computer algebra symbolic and algebraic computation». En: *ACM SIGSAM Bulletin* 16.4 (nov. de 1982), págs. 5-5. URL: <https://doi.org/10.1145/1089310.1089312>.
- [8] C. Cedillo, R. MacKinney-Romero, M. A. Pizaña, I. A. Robles y R. Villarroel-Flores. *YAGS - Yet Another Graph System, Version 0.0.5*. 2019.
- [9] C. Cedillo y M. A. Pizaña. «Clique-divergence is not first-order expressible for the class of finite graphs». En: *Ars Combinatoria* 152 (2020).
- [10] C. Cedillo y M. A. Pizaña. «Clique-convergence is undecidable for automatic graphs». En: *Journal of Graph Theory* 96.3 (2021), págs. 414-437. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/jgt.22622>.
- [11] V. Černý. «Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm». En: *Journal of Optimization Theory and Applications* 45.1 (1985), págs. 41-51. URL: <https://doi.org/10.1007/BF00940812>.
- [12] S. Chatterjee y A. S. Hadi. «Covariance and correlation coefficient». En: *Regression analysis by example*. 5th. Wiley, 2012, 25–30.

- [13] F. Dekking, C. Kraaikamp, L. H.P y L. Meester. «The method of least squares». En: *A modern introduction to probability and statistics: Understanding why and how*. Springer, 2005, 329–336.
- [14] E. Deza y M. M. Deza. *Encyclopedia of Distances*. Springer Berlin Heidelberg, 2009, pág. 192. URL: <https://doi.org/10.1007/978-3-642-00234-2>.
- [15] P. Erdős. «On cliques in graphs». En: *Israel Journal of Mathematics* 4.4 (1966), págs. 233-234. URL: <https://doi.org/10.1007/BF02771637>.
- [16] A. P. Figueroa y B. Llano. «An infinite family of self-diclique digraphs». En: *Appl. Math. Lett.* 23.5 (2010), págs. 630-632. URL: <http://dx.doi.org/10.1016/j.aml.2010.01.026>.
- [17] A. S. Fraser. «Simulation of Genetic Systems by Automatic Digital Computers I. Introduction». En: *Australian Journal of Biological Sciences* 10.4 (1957), págs. 484-491. URL: <https://doi.org/10.1071/B19570484>.
- [18] M. E. Frías-Armenta, F. Larrión, V. Neumann-Lara y M. A. Pizaña. «Local cutpoints and iterated clique graphs». En: *Proceedings of GRA-CO2005*. Vol. 19. Electron. Notes Discrete Math. Amsterdam: Elsevier, 2005, 345-349 (electronic).
- [19] M. E. Frías-Armenta, V. Neumann-Lara y M. A. Pizaña. «Dismantlings and iterated clique graphs». En: *Discrete Math.* 282.1-3 (2004), págs. 263-265.
- [20] M. Frías-Armenta, F. Larrión, V. Neumann-Lara y M. Pizaña. «Edge contraction and edge removal on iterated clique graphs». En: *Discrete Applied Mathematics* 161.10–11 (2013), págs. 1427 -1439.
- [21] R. M. Friedberg. «A Learning Machine: Part I». En: *IBM Journal of Research and Development* 2.1 (ene. de 1958), págs. 2-13. URL: <https://doi.org/10.1147/rd.21.0002>.
- [22] R. M. Friedberg, B. Dunham y J. H. North. «A Learning Machine: Part II». En: *IBM Journal of Research and Development* 3.3 (jul. de 1959), págs. 282-287. URL: <https://doi.org/10.1147/rd.33.0282>.
- [23] GAP – Groups, Algorithms, and Programming, Version 4.11.0. The GAP Group. 2020. URL: <https://www.gap-system.org>.
- [24] D. E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Professional, 1989. URL: <https://www.xarg.org/ref/a/0201157675/>.
- [25] M. E. Groshaus y L. P. Montero. «The number of convergent graphs under the biclique operator with no twin vertices is finite». En: *LAGOS'09—V Latin-American Algorithms, Graphs and Optimization Symposium*. Vol. 35. Electron. Notes Discrete Math. Elsevier Sci. B. V., Amsterdam, 2009, págs. 241-246. URL: <http://dx.doi.org/10.1016/j.endm.2009.11.040>.

- [26] M. Groshaus, A. L. P. Guedes y L. Montero. «Almost every graph is divergent under the biclique operator». En: *Discrete Appl. Math.* 201 (2016), págs. 130-140. URL: <http://dx.doi.org/10.1016/j.dam.2015.07.022>.
- [27] M. Groshaus, P. Hell y J. Stacho. «On edge-sets of bicliques in graphs». En: *Discrete Appl. Math.* 160.18 (2012), págs. 2698-2708. URL: <http://dx.doi.org/10.1016/j.dam.2012.02.004>.
- [28] M. Groshaus y L. P. Montero. «On the iterated biclique operator». En: *J. Graph Theory* 73.2 (2013), págs. 181-190. URL: <http://dx.doi.org/10.1002/jgt.21666>.
- [29] M. Groshaus y J. L. Szwarcfiter. «Biclique graphs and biclique matrices». En: *J. Graph Theory* 63.1 (2010), págs. 1-16. URL: <http://dx.doi.org/10.1002/jgt.20442>.
- [30] A. H. Guth. *Time Since the Beginning*. 2003. arXiv: [astro-ph/0301199](http://arxiv.org/abs/astro-ph/0301199) [astro-ph].
- [31] R. C. Hamelink. «A partial characterization of clique graphs». En: *Journal of Combinatorial Theory* 5.2 (1968), págs. 192 -197. URL: <http://www.sciencedirect.com/science/article/pii/S0021980068800559>.
- [32] R. Hammack, W. Imrich y S. Klavzar. *Handbook of Product Graphs, Second Edition*. 2nd. USA: CRC Press, Inc., 2011.
- [33] S. Hazan y V. Neumann-Lara. «Fixed points of posets and clique graphs». En: *Order* 13.3 (1996), págs. 219-225.
- [34] S. Hedetniemi y P. Slater. «Line Graphs of Triangleless Graphs and Iterated Clique Graphs». En: *Lecture Notes in Mathematics*. Vol. 303. Shaker Heights, Ohio, USA: Springer, 1971, págs. 139-147.
- [35] J. Holland. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. University of Michigan Press, 1975.
- [36] G. Hornby, A. Globus, D. Linden y J. Lohn. «Automated Antenna Design with Evolutionary Algorithms». En: *Space 2006*. American Institute of Aeronautics y Astronautics, sep. de 2006. URL: <https://doi.org/10.2514/6.2006-7242>.
- [37] H. Iba, S. Akiba, T. Higuchi y T. Sato. «BUGS: A Bug-Based Search Strategy using Genetic Algorithms». En: *PPSN*. 1992.
- [38] R. M. Karp. «Reducibility among combinatorial problems». En: *Complexity of Computer Computations*. Ed. por R. Miller y J. Thatcher. Plenum Press, 1972, págs. 85-103.
- [39] F. Larrión y V. Neumann-Lara. «A Family of Clique Divergent Graphs with Linear Growth». En: *Graphs and Combinatorics* 13.3 (1997), págs. 263-266. URL: <https://doi.org/10.1007/BF03353003>.

- [40] F. Larrión, V. Neumann-Lara y M. A. Pizaña. «On expansive graphs». En: *European J. Combin.* 30.2 (2009), págs. 372-379. URL: <http://www.sciencedirect.com/science/article/pii/S0195669808001431>.
- [41] F. Larrión, M. Pizaña y R. Villarroel-Flores. «Iterated clique graphs and bordered compact surfaces». En: *Discrete Mathematics* 313.4 (2013), págs. 508-516.
- [42] F. Larrión, M. Pizaña y R. Villarroel-Flores. «On the clique behavior of circulants with three small jumps». En: *Electronic Notes in Discrete Mathematics* 35 (2009), págs. 341-346. URL: <https://www.sciencedirect.com/science/article/pii/S1571065309002091>.
- [43] R. D. Luce y A. D. Perry. «A method of matrix analysis of group structure». En: *Psychometrika* 14.2 (1949), págs. 95-116. URL: <https://doi.org/10.1007/BF02289146>.
- [44] O. Z. Maimon y D. Braha. «A genetic algorithm approach to scheduling PCBs on a single machine». En: *International Journal of Production Research* 36.3 (1998), págs. 761-784. URL: <https://doi.org/10.1080/002075498193688>.
- [45] E. Mayr. *What evolution is*. London: Phoenix, 2002, págs. 100 -117.
- [46] B. McKay. *Various simple graphs*. 2020. URL: <http://users.cecs.anu.edu.au/~bdm/data/graphs.html> (visitado 01-08-2020).
- [47] R. Miller y D. Muller. «A problem of maximum consistent subsets». En: *IBM Research Report RC-240* (1960).
- [48] J. Moon y L. Moser. «On cliques in graphs». En: *Israel Journal of Mathematics* 3 (1 1965), págs. 23-28.
- [49] V. Neumann-Lara. «On Clique-divergent Graphs». En: *Problèmes combinatoires et théorie des graphes (Colloq. Internat. CNRS, Univ. Orsay, Orsay, 1976)*. Vol. 260. Colloq. Internat. CNRS. Paris: CNRS, 1978, págs. 313-315.
- [50] V. Neumann-Lara. «Clique divergence in graphs». En: *Algebraic methods in graph theory, Coll. Math. Soc. János Bolyai, vol. 25 Szeged*. Ed. por L. Lovász y V. T. Sós. Amsterdam: North-Holland, 1981, págs. 563-569.
- [51] C. Notredame y D. G. Higgins. «SAGA: Sequence Alignment by Genetic Algorithm». En: *Nucleic Acids Research* 24.8 (abr. de 1996), págs. 1515-1524. URL: <https://doi.org/10.1093/nar/24.8.1515>.
- [52] M. Pizaña e I. Robles. «On cliques and bicliques». En: *Electronic Notes in Discrete Mathematics* 62 (2017), págs. 189 -194. URL: <http://www.sciencedirect.com/science/article/pii/S157106531730272X>.
- [53] M. Pizaña e I. Robles. «On bicliques and the second clique graph of suspensions». En: *Discrete Applied Mathematics* 281 (2020), págs. 261 -267. URL: <http://www.sciencedirect.com/science/article/pii/S0166218X19300988>.

- [54] E. Prisner. *Graph dynamics*. Harlow: Longman, 1995.
- [55] M. Requardt. «Space-time as an order-parameter manifold in random networks and the emergence of physical points». En: *Quantum theory and symmetries (Goslar, 1999)*. World Sci. Publ., River Edge, NJ, 2000, págs. 555-561.
- [56] M. Requardt. «A geometric renormalization group in discrete quantum space-time». En: *J. Math. Phys.* 44.12 (2003), págs. 5588-5615.
- [57] F. S. Roberts y J. H. Spencer. «A characterization of clique graphs». En: *Journal of Combinatorial Theory, Series B* 10.2 (1971), págs. 102-108. URL: <http://www.sciencedirect.com/science/article/pii/0095895671900700>.
- [58] D. Schneider. «The Exascale Era is Upon Us: The Frontier supercomputer may be the first to reach 1,000,000,000,000,000,000 operations per second». En: *IEEE Spectrum* 59.1 (2022), págs. 34-35.
- [59] N. Sloane. *Number of graphs on n unlabeled nodes: sequence A000088 in the On-Line Encyclopedia of Integer Sequences*. <https://oeis.org/A000088>.
- [60] J. L. Szwarcfiter. «A survey on clique graphs». En: *Recent advances in algorithms and combinatorics*. Ed. por B. A. Reed y C. Linhares-Sales. Vol. 11. CMS Books Math./Ouvrages Math. SMC. New York: Springer, 2003, págs. 109-136.
- [61] J. L. Szwarcfiter. «Recognizing clique-Helly graphs». En: *Ars Combin.* 45 (1997), págs. 29-32.
- [62] J. T. Troelsen. «Adult-type hypolactasia and regulation of lactase expression». En: *Biochimica et Biophysica Acta (BBA) - General Subjects* 1723.1 (2005), págs. 19-32. URL: <https://www.sciencedirect.com/science/article/pii/S0304416505000346>.
- [63] S. S. Vazhkudai y col. «The Design, Deployment, and Evaluation of the CORAL Pre-Exascale Systems». En: (jul. de 2018).
- [64] M. Whitlock. «Evolutionary Processes». En: *The Princeton guide to evolution*. Ed. por J. Losos. Princeton, United Kingdom: Princeton University Press, 2014. Cap. IV, págs. 305-306.
- [65] D. Wood. «On the number of maximal independent sets in a graph». En: *Discrete Math. Theor. Comput. Sci.* 13.3 (2011), págs. 17-19.
- [66] B. Zelinka. «On a problem of E. Prisner concerning the biclique operator». En: *Math. Bohem.* 127.3 (2002), págs. 371-373.

ÍNDICE ALFABÉTICO

A

algoritmo genético
 codificación en un, 37
 cruzamiento en un, 52
 decodificación en un, 37
 función de aptitud de un, 38
 inicialización de un, 47
 pasos generales de un, 46
 reproducción en un, 47
algoritmo(s)
 evolutivos, 35
 genéticos, 35

B

biclan(es), 11
 gráfica de, 12
 función de, 17
biológico(a)
 alelo(s), 36
 aptitud, 38
 evolución, 35
 gen(es), 36

C

clan
 comportamiento, 2
 convergente, 2
 divergente, 2
conjunto potencia, 3
covarianza, 39

E

estructura de datos
 lista enlazada, 5

G

GAP, 5
 función Sum de , 49
genética

 información, 36
 mutación(es), 53
genético(a)
 entrecruzamiento, 52
grupo de enteros modulo n , 4
gráfica(s)
 $\mathcal{G}_n$, 10
 ciclo, 4
 clan-Helly, 2
 complemento, 4
 completa, 4
 de clanes, 2
 de Miller-Muller, 1
 intersección, 2
 iteradas de clanes, 2
 octaedro, 4
 octaedro d -dimensional, 4
 producto categórico fuerte
 de, 4
 producto tensorial de, 4
 sin aristas, 4
 sin un vértice, 4
 sin vértices, 4
 trayectoria, 4
 unión de, 4
 clan de una, 1
 de Miller-Muller, 9
 isomorfismo de, 3
 orden de una, 3
 producto bola de, 15
 producto tensorial de, 15
 suspensión de una, 11

H

Helly
 propiedad de, 2

M

Miller

R.E., 9

Moon

J.W., 9

Moser

L., 9

Muller

D.E., 9

N

notación

Ω , 4

Θ , 4

O grande, 4

O

operador de clanes K, 2

P

problema

decidible, 3

NP-Completo, 2

problema de optimización

espacio de búsqueda de un,
36

solución candidata de un, 36

programación estructurada, 5

promedio, 39

proteínas, 36

S

selección natural, 38

sistema de álgebra

computacional, 5

T

teorema

del triángulo extendido, 2

V

varianza, 39

vértice(s)

universal, 20

vecindad abierta de un, 3

vecindad cerrada de un, 3

vecindad cerrada de un
conjunto de, 16

vértices(s)

conjunto no-cubrible de, 24

W

Wood

David R., 9

Y

YAGS, 5

función CliqueGraph de , 42

función NumberOfCliques
de , 42

Z

Zykov

suma de, 4