



UNIVERSIDAD DE SONORA

DIVISIÓN DE CIENCIAS EXACTAS Y NATURALES

Programa de Posgrado en Matemáticas

Propiedades homológicas y de colapsibilidad
asociadas a gráficas \mathfrak{J} -contraíbles

T E S I S

Que para obtener el grado académico de:

Doctorado en Ciencias Matemáticas

Presenta:

Héctor Alfredo Hernández Hernández

Directores de tesis:

Dr. Jesús Francisco Espinoza Fierro

Dr. Martín Eduardo Frías Armenta.

Hermosillo, Sonora, México

5 de Febrero de 2019

Universidad de Sonora

Repositorio Institucional UNISON



“El saber de mis hijos
hará mi grandeza”



Excepto si se señala otra cosa, la licencia del ítem se describe como openAccess

Sinodales

Dr. Julio César Ávila Romero

Instituto Tecnológico y de Estudios Superiores de Monterrey

Dr. Jesús Francisco Espinoza Fierro

Departamento de Matemáticas

Universidad de Sonora

Dr. Étienne Fieux

Institut de Mathématiques de Toulouse

Université Paul Sabatier

Dr. Martín Eduardo Frías Armenta

Departamento de Matemáticas

Universidad de Sonora

Dr. Pedro Eduardo Miramontes Vidal

Facultad de Ciencias

Universidad Nacional Autónoma de México

Dedico este trabajo una persona admirable, muy capaz y que ha influido de manera importante en mi formación como matemático:

al Dr. Pedro Flores Pérez.

Agradecimientos

Quiero expresar mi más sincero agradecimiento a todas las personas que han contribuido en la elaboración de este trabajo.

A mi familia:

Por su incomensurable paciencia y apoyo, a mi esposa Paola Tonanzy, a mi comadre Bertha Silvia y a nuestras tres bendiciones Ytrethzy, Fer y nuestro quelite Andrea. A mi cuñado Fco. Rafael y familia, haaaa! y a mi apá, a mi amá y a mis hermanos.

A mis directores de tesis:

Dr. Martín Eduardo Frías y Dr. Jesús Francisco Espinoza, por todo el inmenso apoyo y dedicación al desarrollo de este trabajo, por confiar en mí, y por dejarme ser :).

Al comité revisor que incluye al Dr. Pedro Miramontes, al Dr. Étienne Fioux y al Dr. Julio Ávila, por las atinadas observaciones para el desarrollo del presente y futuro trabajo.

A todos mis amigos maestros y trabajadores, que directamente o indirectamente me han brindado su apoyo, sin olvidar su inmensa ayuda para ponerme en circulación en el mes de noviembre de 2016.

Índice general

Introducción	x
1. Preliminares	1
1.1. Gráficas simples no dirigidas	1
1.2. Subgráficas	2
1.3. Isomorfismos entre gráficas	7
1.4. Forma canónica maximal de una gráfica	9
1.5. Etiquetado canónico	10
1.6. Complejos simpliciales	21
2. Familias de gráficas	23
2.1. Familia de gráficas conexas	23
2.2. Familia de gráficas \mathcal{I} -contraíbles	30
2.3. Gráficas colapsables	36
2.4. Comparación de familias fuertemente \mathcal{I} -contraíbles y colapsables	37
2.5. Construcción iterada de gráficas fuertemente \mathcal{I} -contraíbles	42
2.6. Gráficas homotópicamente colapsables	44
3. Propiedades homológicas de gráficas fuertemente \mathcal{I}-contraíbles	47
3.1. Homología de gráficas	47
3.2. Invariancia homológica del borrado de vértices fuertemente \mathcal{I} -contraíbles	48
3.3. Aplicación: Homología persistente del complejo de Vietoris-Rips	50
4. Conclusiones y trabajo a futuro	55
4.1. Resultados obtenidos	55
4.2. Trabajo a futuro	55
Apéndices	57
A. Ejemplos de gráficas conexas y fuertemente \mathcal{I}-contraíbles	59
A.1. Primeras 50 gráficas conexas	59
A.2. Primeras gráficas fuertemente \mathcal{I} -contraíbles	64
B. Transformación del <i>dunce hat</i> a $K(17)$	67
Bibliografía	73

Resumen

El estudio de las gráficas es muy importante en las aplicaciones de las matemáticas y también para fines teóricos. La primera parte de este trabajo sirve como una introducción a la teoría de gráficas y a los complejos simpliciales. La segunda parte tiene que ver con el efecto de aplicar cierto tipo de reducciones a las gráficas a fin de que el cálculo de las características homológicas de las gráficas sea económico.

También se incluye una explicación detallada de la técnica de Etiquetado Canónico, necesaria en la resolución del problema de isomorfismo entre gráficas.

Al parecer esta es la estrategia planteada es la más recomendable para efectos del cálculo de su homología.

Así este trabajo está dirigido para quienes desean introducirse a la teoría de gráficas, complejos simpliciales y a quienes deseen empezar a estudiar Análisis Topológico de Datos.

Introducción

En [15, 16], A. Ivashchenko muestra una familia de gráficas construida a partir de $K(1)$ por medio de transformaciones contraíbles (como en la Definición 2.2.1), él prueba que tales transformaciones no cambian los grupos de homología de una gráfica. A. Ivashchenko comenzó el estudio de estas transformaciones porque las usó en teoría de espacios moleculares y topología digital. Las referencias modernas son [6, 13].

En [5] se estudian *ws*-desmantelabilidad y colapsabilidad de una gráfica; ver Observación 2.4.5. Una gráfica es colapsable si su complejo de subgráficas completas es colapsable (ver Sección 2.3.). Ninguna de las dos operaciones anteriores (*ws*-desmantelabilidad y colapso por una cara libre) cambia los grupos de homología de una gráfica (ver [5]).

En este trabajo se estudia la relación entre gráficas contraíbles en el sentido de Ivashchenko y la familia de gráficas colapsables, el resultado principal es utilizado para disminuir la complejidad en los cálculos de los grupos de homología en el complejo de Vietori-Rips. Adicionalmente, se obtienen resultados utilizables en el estudio de las gráficas simples.

En el Capítulo 1 introducimos los conceptos básicos, a fin de crear un documento autocontenido en la medida de lo posible, de esta parte vale la pena destacar el método para resolver el problema de isomorfismo entre gráficas simples con un número reducido de vértices.

En el Capítulo 2 se presenta un algoritmo para la construcción de la familia de gráficas conexas, consideramos que sugiere las bases para la construcción de un estándar para la familia de gráficas conexas. También se presenta la familia de gráficas contraíbles y fuertemente contraíbles en el sentido de Ivashchenko. Se presenta la familia de gráficas colapsables y se elaboran algoritmos para construir colecciones de éstas dos familias.

En la Sección 2.3, comenzamos con alguna terminología y notación de complejos simpliciales y damos la definición de gráfica colapsable en términos de la colapsabilidad del complejo de subgráficas completas. Mas aún, probamos que cualquier gráfica fuertemente contraíble (en el sentido de Ivashchenko) es también una gráfica colapsable: Teorema 2.4.1. Como una consecuencia, los grupos de homología de cualquier gráfica contraíble son triviales, como se probó en [15].

Conjeturamos que cualquier gráfica colapsable es también una gráfica fuertemente contraíble (en el sentido de Ivashchenko). Sin embargo, no podemos seguir argumentos similares como los usados en el Teorema 2.4.1. Comenzamos la Sección 2.4 con un ejemplo que muestra que el Teorema 2.4.1 no es trivialmente reversible.

A través de algoritmos mostramos evidencia computacional que apoya la conjetura. Los códigos fueron escritos en C/C++, son de código abierto y están disponibles en el Repositorio [3] en <http://www.gcs.mat.uson.mx/>.

También se muestra la necesidad de ampliar a familia de gráficas colapsables. De hecho conjeturamos que la familia de gráficas contraíbles en el sentido de Ivashchenko, coincide con la familia ampliada de gráficas colapsables.

En el Capítulo 3 introducimos el concepto de homología de una gráfica como en [15] y las transformaciones contraíbles como fueron presentadas en [16]. Concluimos la sección con el Lema 3.2.2, para mostrar la existencia de un homomorfismo inducido entre grupos de homología de dos gráficas, dado por la imagen de transformaciones contraíbles.

Concluimos la Sección 3.3 con una aplicación de este trabajo al cálculo de homología persistente del complejo de Vietoris-Rips. Este es un tema de interés en análisis topológico de datos.

En el Capítulo 4 se muestra un listado de resultados obtenidos y del trabajo que se visualiza en el mediano plazo.

Finalmente, en el Apéndice A se muestra evidencia principalmente gráfica que el lector puede consultar, evita saturar el escrito principal. El Apéndice B contiene la sucesión de pasos que demuestra que la gráfica asociada a cierta triangulación de la superficie conocida como *dunce hat* es contraíble, pero no fuertemente contraíble (en el sentido de Ivashchenko).

Capítulo 1

Preliminares

En este capítulo se presentan brevemente un resumen conceptos elementales requeridos en este trabajo, mostrando ejemplos sencillos, así como algunas de las formas en las que una gráfica simple puede ser presentada.

1.1. Gráficas simples no dirigidas

Probablemente las gráficas simples es uno de los conceptos matemáticos más apropiados para ser utilizados para indicar si entre elementos de un conjunto finito existe o no cierta relación.

Definición 1.1.1. Una gráfica simple no dirigida G está determinada por un conjunto finito de vértices V y un conjunto de aristas E

$$G = (V, E)$$

donde $V = \{v_1, v_2, \dots, v_n\}$ y $E \subseteq \{\{v_i, v_j\} \mid v_i, v_j \in V, \text{ con } i \neq j\}$.

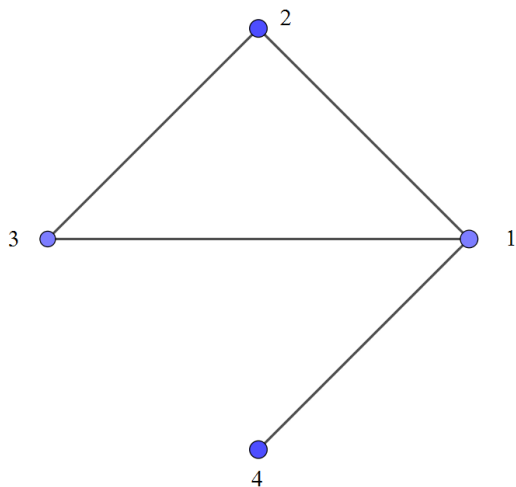
En este trabajo nos referiremos de aquí en adelante a las gráficas simples sólo como **gráficas** y nos restringimos a gráficas finitas.

Observaciones. 1.1.2.

- Es frecuente utilizar números naturales para denotar o etiquetar a cada vértice v_i .
- El término “simple” se refiere que a lo más existe una arista entre cada par de vértices y a que no existe orientación de la arista.
- En este tipo de gráficas no existen aristas de un vértice a si mismo, es decir, no existen lazos.

Representación geométrica. Los vértices V se representan por puntos y las aristas E se representan por segmentos de línea entre los puntos. La colocación de los vértices es arbitraria, pero se busca que sea evidente qué vértices une cada segmento. Una forma que cumple con la característica mencionada es colocar vértices alrededor de un círculo de manera homogénea, en varias colecciones de representaciones geométricas de gráficas esto es común.

Ejemplo 1.1.3. Sea $G = (V, E)$ la gráfica con $V = \{1, 2, 3, 4\}$ y $E = \{\{1, 2\}, \{1, 3\}, \{1, 4\}, \{2, 3\}\}$. Su representación geométrica es:



Representación matricial. Una gráfica se puede representar por medio de una matriz $M = (a_{i,j})$ de la siguiente manera:

$$a_{i,j} = \begin{cases} 1 & \text{si } \{v_i, v_j\} \in E, \\ 0 & \text{si } \{v_i, v_j\} \notin E. \end{cases}$$

A la matriz M se le llama *matriz de adyacencia*.

Por ejemplo, la representación matricial de la gráfica anterior es:

$$M = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}.$$

Observaciones. 1.1.4.

- Las entradas de la matriz de adyacencia son elementos del conjunto $\{0, 1\}$, no se asignan pesos a las aristas, sólo se indica si existe o no la arista en la gráfica.
- La matriz de adyacencia es simétrica. La arista $\{v_i, v_j\}$ también se puede expresar como $\{v_j, v_i\}$, ambas expresiones se registran en la matriz.
- La diagonal principal sólo contiene ceros, pues no existen lazos.

1.2. Subgráficas

En teoría de gráficas es común que centremos la atención en partes de una gráfica, que a su vez son una gráfica.

Definición 1.2.1. Se dice que la gráfica $G' = (V', E')$ es **subgráfica** de $G = (V, E)$ si $V' \subset V$ y $E' \subset E$. Si E' es maximal, entonces G' es llamada **subgráfica inducida** por V' .

Que E' sea maximal significa que para toda arista $\{v_1, v_2\} \in E$ tal que $v_1, v_2 \in V'$ se tiene que $\{v_1, v_2\} \in E'$.

Para obtener la matriz de adyacencia de la subgráfica inducida por los vértices V' , G' basta extraer los renglones y columnas de la matriz M correspondientes a los vértices V' .

Ejemplo 1.2.2. Sean $V = \{1, 2, 3, 4, 5\}$, $V' = \{1, 2, 4\}$ y

$$E = \{\{1, 2\}, \{1, 3\}, \{1, 4\}, \{1, 5\}, \{2, 4\}, \{2, 5\}, \{3, 4\}, \{4, 5\}\}.$$

Las matrices de adyacencia correspondientes son:

		1	2	3	4	5
	1	0	1	1	1	1
	2	1	0	0	1	1
	3	1	0	0	1	0
	4	1	1	1	0	1
	5	1	1	0	1	0

 $M =$

		1	2	4
	1	0	1	1
	2	0	0	1
	4	1	1	0

 $M' =$

Lo anterior se logra utilizando composición de índices, donde $a'_{i,j} = a_{I_i, I_j}$, $i, j = 1, 2, 3$, con $I = (1, 2, 4)$.

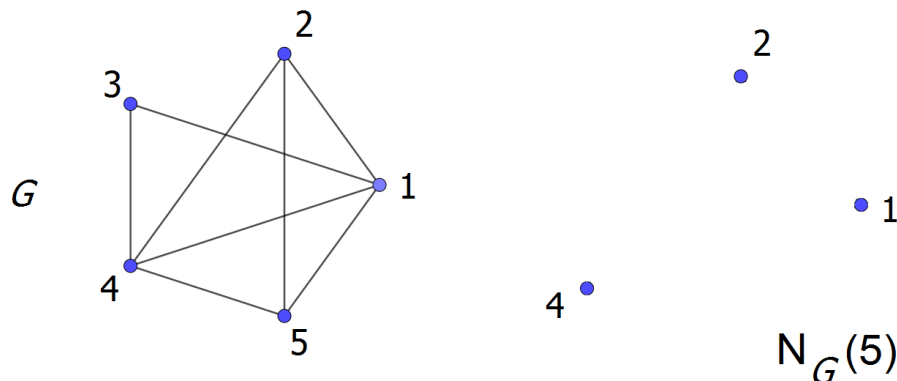
Existen varios tipos de subgráficas que valen la pena destacar.

Definición 1.2.3. Sea $G = (V, E)$ y $v \in V$, la **vecindad abierta** del vértice v es

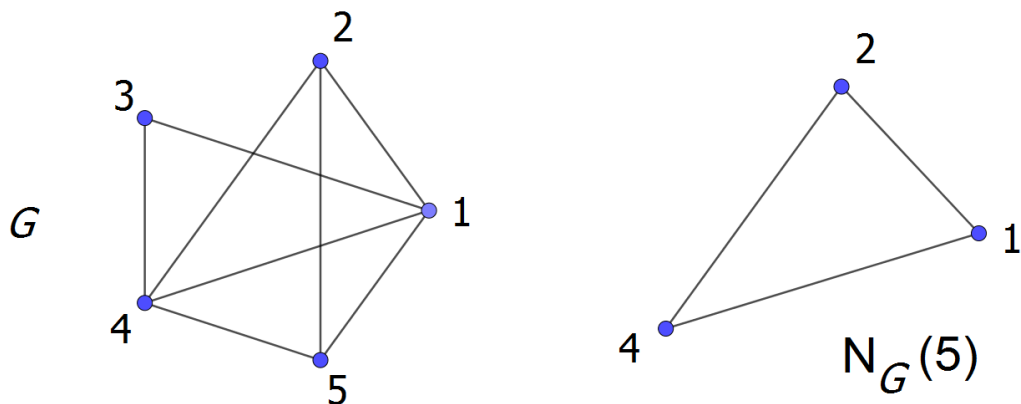
$$N_G(v) = \{u \in V \mid \{u, v\} \in E, u \neq v\}.$$

Cuando el contexto lo permite, denotamos a la vecindad abierta por $N(v)$.

Ejemplo 1.2.4. Para la gráfica G de la siguiente figura, la vecindad abierta del vértice 5 es $N_G(5)$ es:



Observación. 1.2.5. Abusando un poco de la notación y cuando no hay lugar a confusión se utilizará $N_G(v)$ o $N(v)$ para referirse a la subgráfica inducida por los vértices vecinos del vértice v , esto es:

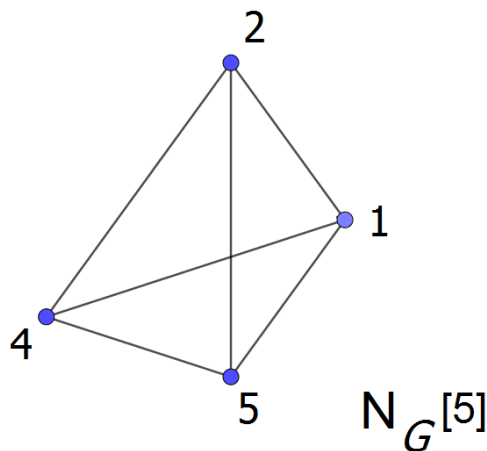


La representación matricial de esta vecindad está dada en el Ejemplo 1.2.2.

Dada una gráfica $G = (V, E)$, se define la **vecindad cerrada** de $v \in V$ como:

$$N_G[v] = N_G(v) \cup \{v\}.$$

También se podrá denotar por $N[v]$ cuando no haya riesgo de confusión respecto a la gráfica G . Por ejemplo, para la gráfica anterior la subgráfica $N_G[5]$ es:



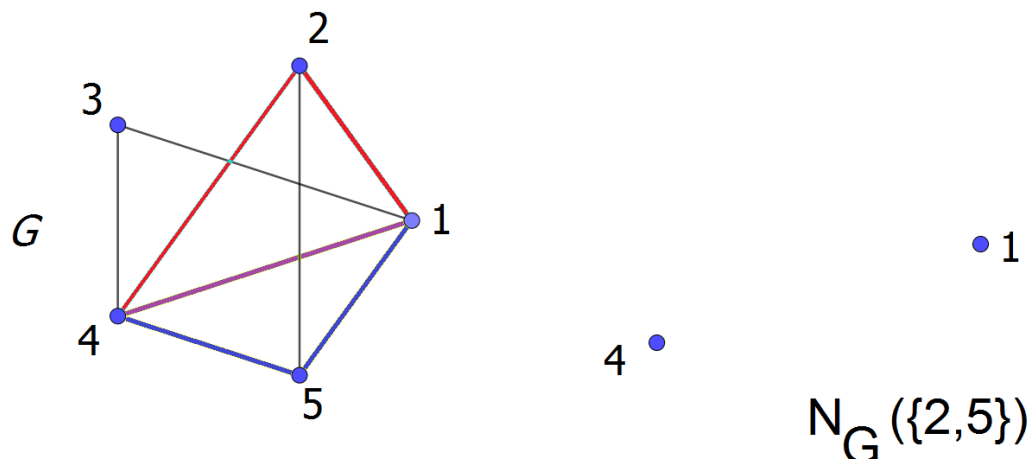
Definición 1.2.6. Sea $G = (V, E)$ una gráfica. La **vecindad abierta de una arista** $\{x, y\} \in E$ se define como:

$$N_G(\{x, y\}) = N_G(x) \cap N_G(y).$$

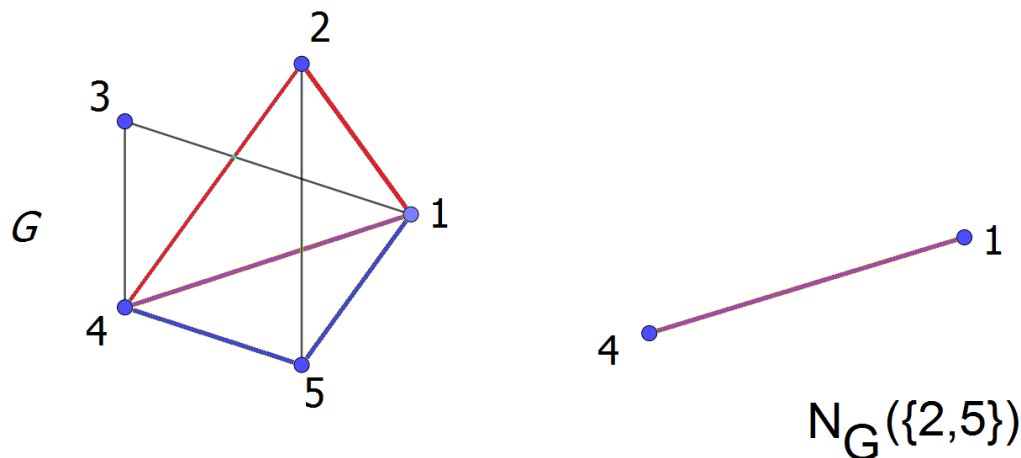
Ejemplo 1.2.7. Consideremos la gráfica G dada en el Ejemplo 1.2.4, la vecindad abierta de la arista $\{2, 5\} \in E$ es:

$$N_G(\{2, 5\}) = N_G(2) \cap N_G(5) = \{1, 4, 5\} \cap \{1, 2, 4\} = \{1, 4\}.$$

La vecindad anterior consiste de un par de vértices, sin la arista entre ellos.



De la misma manera que la Definición 1.2.3, también abusando de la notación, se utilizará $N_G(\{x, y\})$ para referirse a la gráfica inducida por la vecindad abierta de $\{x, y\} \in E$, ejemplo:



Una **gráfica camino** está determinada por una secuencia de vértices, con la condición de que siempre exista una arista entre un vértice y el siguiente, i.e.:

$C = (V', E')$ con $V' = \{v_{i_1}, v_{i_2}, \dots, v_{i_k}\}$ y $E' = \{\{v_{i_1}, v_{i_2}\}, \{v_{i_2}, v_{i_3}\}, \{v_{i_3}, v_{i_4}\}, \dots, \{v_{i_{k-1}}, v_{i_k}\}\}$ donde todos los vértices $v_{i_1}, v_{i_2}, v_{i_3}, \dots, v_{i_{k-1}}$ son diferentes entre si. Al número de aristas se le llama **longitud del camino**.

Si una gráfica camino termina en el vértice que inicia ($v_{i_1} = v_{i_k}$), se le llama **ciclo** y al número de aristas se le llama la **longitud del ciclo**.

Se define el **grado** del vértice v como el número de aristas de las cuales v forma parte; o dicho de otra manera, el número de vértices vecinos de v .

Dada la matriz de adyacencia de la gráfica G , el grado del vértice v_i se calcula sumando los elementos del renglón i de M , i.e. :

$$\text{grado}(v_i) = \sum_{j=1}^n a_{i,j}$$

Si v_j tiene grado cero, entonces v_j es un vértice aislado.

En una gráfica camino, que no sea un ciclo, el primer y último vértice tienen grado uno y el resto tienen grado dos. En un ciclo todos los vértices son de grado dos.

Si entre cada par de vértices de una gráfica no trivial siempre existe una subgráfica camino, entonces se le llama **gráfica conexa**. Se puede verificar que $\text{grado}(v) \geq 1$ para todo vértice $v \in V$. En $G = K(1)$ no existen pares de vértices, se considera conexa pues consta de “una sola parte”.

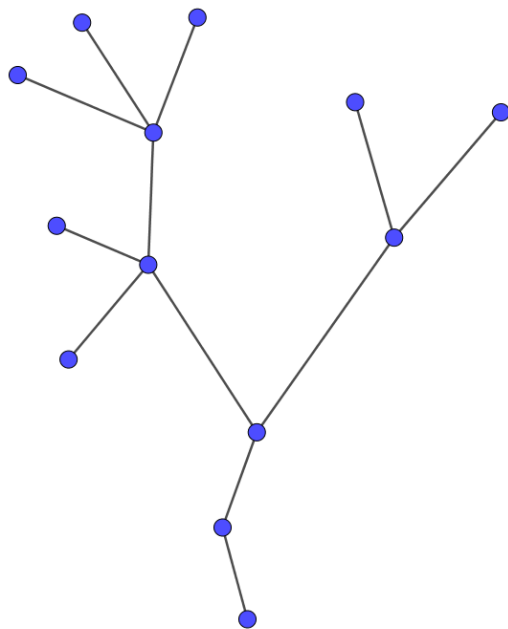
De las gráficas conexas destaca la siguiente:

A una gráfica conexa sin ciclos se le llama **árbol**. Cuando un árbol tiene un número finito de vértices, una manera sencilla de caracterizarlo es como una gráfica conexa que cumple la siguiente relación entre la cardinalidad del conjunto de vértices y la cardinalidad del conjunto de aristas:

$$|V| = |E| + 1.$$

Es fácil verificar que toda gráfica conexa posee una subgráfica árbol. A los vértices de grado uno, junto con la arista que incide en ellos, se les llama **hojas**.

En cierto sentido, un árbol es una subgráfica mínima que “mantiene unido” al conjunto V .

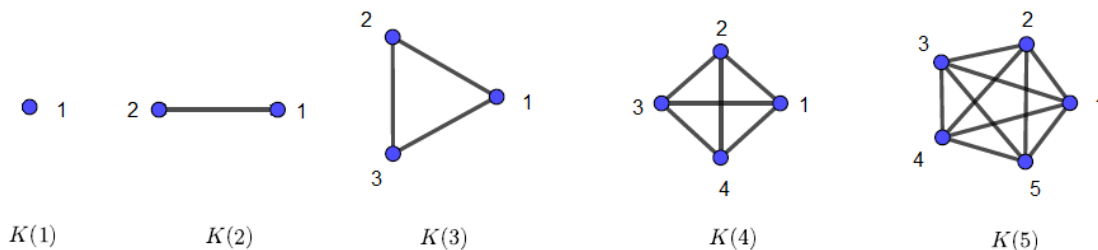


Árbol con $|V|=13$ y $|E|=12$.

Por otra parte, a la gráfica máxima, la que posee todas las aristas posibles se le llama **gráfica completa** y cumple la siguiente propiedad:

$$|E| = \frac{|V|(|V| - 1)}{2}.$$

En una gráfica completa todos los vértices tienen grado $|V| - 1$. Se utiliza $K(n)$ para denotar a la gráfica completa de n vértices, enseguida se muestran cinco representaciones geométricas de gráficas completas



Sus nombres son: vértice, arista, triángulo, tetraedro, $K(5)$, en general la dimensión de $K(n)$ es $n - 1$.

1.3. Isomorfismos entre gráficas

El siguiente concepto es fundamental para generar familias de gráficas. Contar con colecciones amplias de gráficas suele ser necesario para probar algoritmos, descartar o validar conjeturas. Que estos bancos o familias de gráficas sólo incluyan gráficas esencialmente diferentes unas de otras es muy importante, pues evita en muchos casos realizar trabajo innecesario.

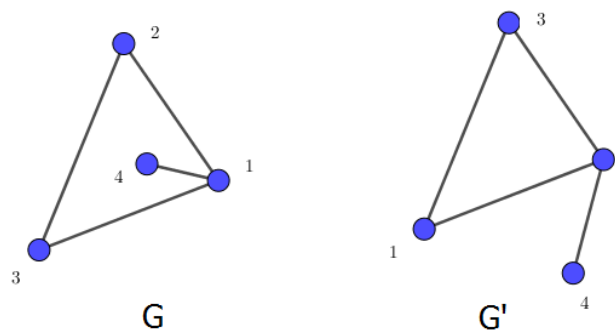
Definición 1.3.1. Se dice que dos gráficas $G = (V, E)$ y $G' = (V', E')$ son **isomorfas**, si existe una función biyectiva $\pi : V \rightarrow V'$ tal que

$$\{v_i, v_j\} \in E \iff \{\pi(v_i), \pi(v_j)\} \in E'.$$

Cuando se tiene que $V = V'$ son finitos, estas funciones biyectivas son permutaciones entre índices.

Ejemplo 1.3.2. Sea $G = (V, E)$, con $V = \{1, 2, 3, 4\}$ y $E = \{\{1, 2\}, \{1, 3\}, \{2, 3\}, \{4, 1\}\}$ y sea $G' = (V', E')$, con $V' = V$ y $E' = \{\{1, 2\}, \{1, 3\}, \{2, 3\}, \{4, 2\}\}$.

Las respectivas representaciones geométricas son:



Capítulo 1

Nótese que hay gran similitud entre las representaciones geométricas de gráficas: mismo número de vértices, mismo número de aristas, misma cantidad de vértices de cada grado. Las gráficas G y G' son isomorfas, donde el isomorfismo $\pi : V \rightarrow V'$ está dado por la siguiente permutación:

$$\pi = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 1 & 4 \end{pmatrix}^1.$$

De hecho, otra manera equivalente de definir un isomorfismo entre dos gráficas es la siguiente:

Dos gráficas $G = (V, E)$ y $G' = (V', E')$, se dicen isomorfas ($G' \simeq G$) si existe π permutación de índices tal que

$$G' = \pi(G).$$

Sean M y P las matrices de adyacencia de las gráficas G y $\pi(G)$ respectivamente con n vértices. Entonces la matriz P es una permutación de la matriz M

$$P = \pi(M)$$

y sus entradas se calculan:

$$p_{i,j} = a_{\pi(i),\pi(j)} \quad \text{para } i, j = 1, 2, \dots, n. \quad (1.1)$$

Sean G y G' gráficas isomorfas arbitrarias, toda función f definida sobre las gráficas se dice invariante bajo isomorfismo, si $f(G) = f(G')$. Existen funciones de este tipo muy evidentes:

- Número de vértices.
- Número de aristas.
- Número de componentes conexas.
- Grado máximo y grado mínimo de los vértices.
- Distribución de grados.
- Número de ciclos.
- Etc...

Una manera de probar que dos gráficas G y G' no son isomorfas es proporcionar un invariante f tal $f(G) \neq f(G')$. Es deseable que el cálculo de este tipo de funciones sea de baja complejidad computacional.

¹La permutación

$$\pi_2 = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 3 & 4 \end{pmatrix}$$

es más sencilla, y también exhibe el isomorfismo entre las gráficas dadas.

En caso de que la aplicación de un conjunto de invariantes sea suficiente para determinar el isomorfismo entre gráficas, el conjunto recibe el nombre de **invariante completo**.

Hasta el momento no se reconoce la existencia de un **invariante bajo isomorfismo completo** de bajo costo computacional. Sin embargo existe al menos un invariante bajo isomorfismo completo que consta de una única función f . Los experimentos lo muestran como de baja complejidad computacional, se aborda más adelante.

Por otra parte es común que una gráfica presente simetrías, esto se refleja con permutaciones que dejan invariante la matriz de adyacencia M , incluso la representación geométrica y al conjunto E . Cuando una gráfica es isomorfa a si misma se dice que G es **automorfa**, en símbolos

$$G \simeq G \Leftrightarrow \exists \pi \mid G = \pi(G).$$

Al automorfismo se le identifica con la permutación asociada π . Cuando π es la permutación identidad, al automorfismo se le llama **automorfismo trivial**.

1.4. Forma canónica maximal de una gráfica

Dado que una gráfica puede adquirir varias formas de ser presentada, se hace necesario elegir una sola manera a fin de identificar gráficas que en cierto sentido tiene la misma estructura, forma o conectividad de sus elementos. La forma canónica nos ayuda a resolver este problema.

Dada una gráfica G , es posible construir una cadena de caracteres que la represente, considerando una parte de la matriz de adyacencia asociada a la gráfica. En particular, es posible obtener una cadena de dígitos binarios $\{0, 1\}$ obtenidos de la parte triangular superior de dicha matriz.

Definición 1.4.1. Sea $G = (V, E)$ una gráfica, cuyo conjunto de vértices está totalmente ordenado $V = \{v_1 < \dots < v_n\}$. Si M es la matriz de adyacencia de G , se define el **número de M** como:

$$\text{Num}(M) = a_{1,2} a_{1,3} \dots a_{1,n} a_{2,3} \dots a_{2,n} a_{3,4} \dots a_{3,n} \dots \dots a_{n-1,n} \quad (1.2)$$

donde

$$M = \begin{bmatrix} 0 & a_{1,2} & a_{1,3} & \dots & a_{1,n-1} & a_{1,n} \\ a_{1,2} & 0 & a_{2,3} & \dots & a_{2,n-1} & a_{2,n} \\ a_{1,3} & a_{2,3} & 0 & \dots & a_{3,n-1} & a_{3,n} \\ \vdots & \vdots & \vdots & \vdots & \ddots & 0 \\ a_{1,n} & a_{2,n} & a_{3,n} & \dots & a_{n-1,n} & 0 \end{bmatrix}$$

Nótese que $\text{Num}(M)$ es una cadena formada por $n(n-1)/2$ dígitos binarios que puede ser considerado como un número en sistema binario, o como una *palabra*, incluso como número en sistema binario puede ser expresado en otras bases de numeración. Puede ser convertido a base decimal o hexadecimal, en esta última base la palabra generada es más corta y puede ser más conveniente para efectos de almacenamiento.

Si aplicamos todas las posibles permutaciones a la matriz M , obtenemos $n!$ de números de G , uno por cada permutación, no necesariamente todos diferentes. A los que aprovechamos para definir la siguiente forma estándar.

Definición 1.4.2. La **forma canónica** de la matriz de adyacencia M de una gráfica totalmente ordenada G , se define como:

$$\text{FC}(M) = \pi(M) \text{ tal que } \text{Num}(\pi(M)) \text{ es máximo}$$

donde $\pi : V \rightarrow V$ es biyectiva, las entradas de $\pi(M)$ se calculan como se indicó en la Fórmula 1.1, denotaremos por $\text{NumMax}(M)$ el número $\text{Num}(\text{FC}(M))$.

En otras palabras la forma canónica de una matriz de adyacencia M es la permutación de M , $\pi(M)$ con $\text{Num}(\pi(M))$ máximo. Evidentemente este máximo existe y es único. En caso de que existan automorfismos actuando en G , π no es única.

Observaciones. 1.4.3.

- Tanto $\text{FC}(M)$ como $\text{NumMax}(M)$ son invariantes completos.
- Dada $\text{FC}(M)$ se puede extraer directamente $\text{NumMax}(M)$ y dado $\text{NumMax}(M)$ se puede reconstruir $\text{FC}(M)$.
- Las primeras versiones de etiquetamiento canónico consideran el número mínimo en lugar del número máximo, pues eligen el orden lexicográfico (ver [20]). Aunque en principio parecen indistintos a fin de establecer un estándar, la reconstrucción de la forma canónica a partir de este número mínimo tiene inconvenientes.

De esta manera para decidir si dos gráficas son isomorfas basta comparar sus formas canónicas o los números máximos asociados.

1.5. Etiquetado canónico

Para cantidades reducidas de vértices, es factible obtener la forma canónica de la matriz de adyacencia M examinando cada una de las permutaciones $\pi(M)$ donde se elige $\pi^*(M)$ con $\text{Num}(\pi^*(M))$ mayor. La idea principal de la técnica de etiquetado canónico es precisamente evitar calcular y examinar las $n!$ permutaciones de M .

La técnica de etiquetamiento canónico funciona por etapas; en la última etapa se hace la mejor elección de los primeros índices a fin de que los primeros dígitos de $\text{Num}(M)$ sean los más apropiados para que $\text{Num}(M)$ sea mayor, en la penúltima etapa se refina la elección de índices para garantizar que se siga contruyendo un $\text{Num}(M)$ mayor, dado que ya se hizo lo mejor en la etapa anterior y así sucesivamente hasta llegar a la primera etapa. A este tipo de algoritmos se les reconoce como **backtracking**.

En esta parte se considerará que las gráficas son simples, conexas y con número de vértices $n \geq 2$. La forma canónica de M se va construyendo poco a poco, empezando por los primeros renglones. A medida que avanza el proceso, se van descartando una gran cantidad de permutaciones hasta que queda solamente una.

En este procedimiento se va construyendo un árbol multinivel, el nodo raíz es el conjunto $\{1, 2, \dots, n\}$ que representa a las $n!$ posibles permutaciones. Veamos como se

construyen los nodos de nivel uno.

Sea R_i la cadena de elementos del renglón i de la parte triangular superior en la matriz de adyacencia de la gráfica G , esto es:

$$\begin{aligned} R_1 &= a_{1,2} a_{1,3} \dots a_{1,n} \\ R_2 &= a_{2,3} \dots a_{2,n} \\ R_3 &= a_{3,4} \dots a_{3,n} \\ &\vdots \\ R_{n-1} &= a_{n-1,n} \end{aligned}$$

Al concatenar las cadenas y considerar la notación posicional obtenemos el siguiente número en sistema binario

$$\text{Num}(M) = R_1 R_2 \dots R_{n-1}.$$

Para que $\text{Num}(M)$ sea máximo es necesario que R_1 sea máximo, lo cual se logra eligiendo como posibles primer vértice a los vértices de grado mayor (gm) y a los siguientes vértices, a vértices vecinos del primer vértice.

Sea $\mathbf{w} = (w_1, w_2, \dots, w_{k_1})$ el vector de vértices que coinciden en tener el grado mayor gm , así los k_1 nodos del primer nivel tendrán la siguiente partición de índices:

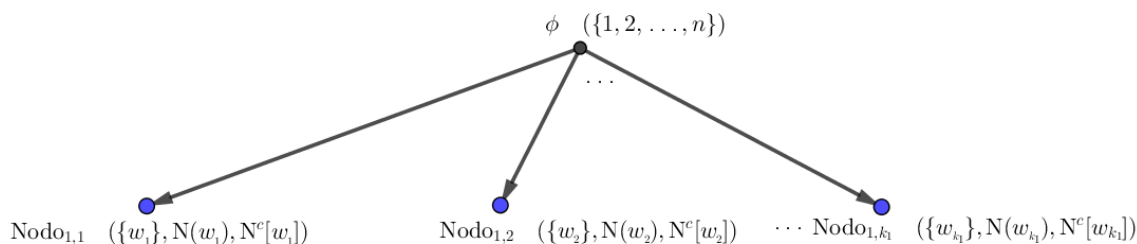
$$(w_i, N(w_i), N^c[w_i]) \quad \text{para } i = 1, 2, \dots, k_1$$

de tal manera que, en todos estos nodos, $R_1 = \underbrace{11 \dots 1}_{gm} \underbrace{00 \dots 0}_{n-gm-1}$ consta de gm unos y $n-gm-1$ ceros.

Observaciones. 1.5.1.

- Nótese que se han colocado la mayor cantidad de unos a la izquierda a fin de que R_1 se máximo, de hecho R_1 tiene $gm = |N(w_i)|$ unos al principio para todo $i \in \{1, 2, \dots, k_1\}$.
- En cada nodo el primer subconjunto es unitario y está fijo, a saber $\{w_i\}$.
- Los vértices $v \in N(w_i)$ pueden permutar entre ellos sin afectar a R_1 , sólo se intercambian unos entre sí.
- Los vértices $v \in N^c[w_i]$ pueden permutar entre ellos sin afectar a R_1 , sólo se intercambian ceros entre sí.
- Cabe la posibilidad de que algunos subconjuntos $N^c[w_i]$ sean vacíos.

Hasta aquí, el árbol construido es el siguiente:



Capítulo 1

En el nivel uno, todos los vértices generan el mismo R_1 , y en principio podríamos conservar todos los nodos, sin embargo, si dos nodos tienen asociada la misma matriz M , no es necesario conservar ambos, uno de ellos puede ser eliminado. Cuando esto sucede estamos detectando un automorfismo no trivial, esto es, estamos ante dos permutaciones $\pi_1 \neq \pi_2$ tales que $\pi_1(M) = \pi_2(M)$, lo que implica que $\pi_2^{-1}\pi_1(M) = M$.

Si algún nodo se elimina, actualizamos el valor de k_1 , de tal manera que tendremos k_1 nodos sobrevivientes del primer nivel, en caso de que algún nodo se elimine se realiza un recorrido de índices.

En la generación de los siguientes niveles del árbol aparecen ligeras diferencias con el primer nivel.

Veamos cómo generar los nodos del segundo nivel, de tal manera que R_1 se conserve y obtengamos R_2 máximo.

Consideremos el $\text{Nodo}_{1,i}$, de este nodo se ramifican $|\text{N}(w_i)|$ nodos de la siguiente manera:

Los candidatos a segundo vértice son todos los $v_2 \in \text{N}(w_i)$, a fin de construir el mejor R_2 para el j -ésimo nodo del nivel dos que se ramifica del nodo i -ésimo de nivel uno, refinamos la partición de la siguiente manera:

$$\text{Nodo}_{2,i_j}$$

$$(\{w_i\}, \{v_{2,i_j}\}, \text{N}(w_i) \cap \text{N}(v_{2,i_j}), \text{N}(w_i) \cap \text{N}^c[v_{2,i_j}], \text{N}^c[w_i] \cap \text{N}(v_{2,i_j}), \text{N}^c[w_i] \cap \text{N}^c[v_{2,i_j}])$$

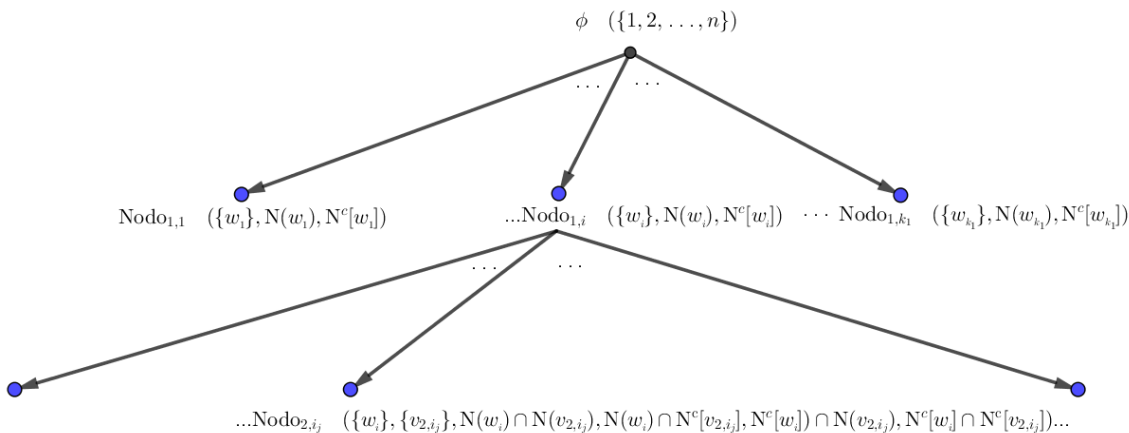
Así, R_2 adquiere la forma

$$\underbrace{11\dots 1}_{|\text{N}(w_i) \cap \text{N}(v_{2,i_j})|} \quad \underbrace{00\dots 0}_{|\text{N}(w_i) \cap \text{N}^c[v_{2,i_j}]|} \quad \underbrace{11\dots 1}_{|\text{N}^c[w_i] \cap \text{N}(v_{2,i_j})|} \quad \underbrace{00\dots 0}_{|\text{N}^c[w_i] \cap \text{N}^c[v_{2,i_j}]|} \quad ,$$

al principio, aparecen $|\text{N}(w_i) \cap \text{N}(v_{2,i_j})|$ unos, luego $|\text{N}(w_i) \cap \text{N}^c[v_{2,i_j}]|$ ceros, después $|\text{N}^c[w_i] \cap \text{N}(v_{2,i_j})|$ unos y por último aparecen $|\text{N}^c[w_i] \cap \text{N}^c[v_{2,i_j}]|$ ceros. Estos últimos vértices son los no vecinos del primer vértice y que tampoco son vecinos del segundo vértice.

En principio eliminaremos a los nodos que no tienen R_2 máximo. Si además, un conjunto de nodos tiene asociada la misma matriz de adyacencia, basta que conservemos sólo un nodo de ese conjunto. Una vez más estaríamos detectando automorfismos no triviales.

Así, el árbol de tres niveles nos queda de la siguiente manera:



Algunos subconjuntos obtenidos de la refinación podrían ser vacíos, en el caso de ser unitarios, no podrán ser particionados en las próximas ramificaciones.

A fin de simplificar la notación de una de las ramificaciones del nodo i de nivel uno, expresamos al j -ésimo nodo del segundo nivel como:

$$(\{w_i\}, \{v_{2,i_j}\}, N_{1,i_j}, N_{2,i_j}, N_{3,i_j}, N_{4,i_j}),$$

con

- $N_{1,i_j} = N(w_i) \cap N(v_{2,i_j})$.
- $N_{2,i_j} = N(w_i) \cap N^c[v_{2,i_j}]$.
- $N_{3,i_j} = N^c[w_i] \cap N(v_{2,i_j})$.
- $N_{4,i_j} = N^c[w_i] \cap N^c[v_{2,i_j}]$.

En general, un nodo de nivel k tiene la forma:

$$(\{v_{i_1}\}, \{v_{i_2}\} \dots \{v_{i_k}\}, N_1, N_2, \dots, N_r)^2, \text{ donde } \{v_{i_1}\} = \{w_i\},$$

y la generación de un nodo de nivel $k+1$ se obtiene refinando la partición de un nodo del nivel anterior, tal refinación tiene la forma

$$(\{v_{i_1}\}, \{v_{i_2}\} \dots \{v_{i_k}\}, \{v_{i_{k+1}}\}, N_1 \cap N(v_{i_{k+1}}), N_1 \cap N^c[v_{i_{k+1}}], \dots, N_r \cap N(v_{i_{k+1}}), N_r \cap N^c[v_{i_{k+1}}]),$$

donde $v_{i_{k+1}} \in N_1$.

Esta manera de refinar la partición garantiza la obtención del máximo R_{k+1} para esa elección de $v_{i_{k+1}}$ sin modificar ninguno de los R_i anteriores.

Nótese que el $(k+1)$ -ésimo subconjunto de la partición se segmenta hasta en tres subconjuntos, a saber

$$\{v_{i_{k+1}}\}, N_1 \cap N(v_{i_{k+1}}), N_1 \cap N^c[v_{i_{k+1}}],$$

con $v_{i_{k+1}} \in N_1$ y del $(k+2)$ -ésimo subconjunto de la partición en adelante se segmentan hasta en dos subconjuntos, como por ejemplo:

$$N_2 \cap N(v_{i_{k+1}}), N_2 \cap N^c[v_{i_{k+1}}].$$

Puede suceder que alguno de ellos sea unitario y el otro vacío (lo cual significa que no se refinó el subconjunto), también puede suceder que ambos subconjuntos sean no unitarios.

Este proceso de etiquetado canónico termina cuando obtenemos un único nodo con todos los subconjuntos unitarios. El árbol generado no tendrá más de $n-1$ niveles, sin contar el nivel cero (nodo raíz).

A continuación se muestra un ejemplo pequeño, se ilustra paso a paso el procedimiento. Se podrá apreciar que se obtienen algunos subconjuntos N_i unitarios antes de que les

²Para simplificar aun más, dejamos de especificar de donde proviene a detalle ese nodo, al omitir el índice i_j .

llegue el turno de ser particionados.

Cada nodo se mostrará junto la permutación de la matriz M correspondiente. El fin es mostrar que es posible eliminar algunos nodos antes de ser calculados, eficientando con ello al proceso.

Nos referiremos a la idea anterior como una **selección preliminar de nodos a ramificar**, ésta consiste es comparar el primer bloque de dígitos que componen a cada R_i a fin de poder descartar posibles ramificaciones, sin necesidad de calcular y comparar R_i por completo, la comparación de los dígitos del primer bloque de R_i , es através de las suma de sus dígitos pues el nuevo nodo aun no ha sido calculado. De hecho se realizan varias sumas, estas corresponden a los renglones de una submatriz cuyos índices son los elementos de N_1 .

Es importante afirmar que esta preselección generaliza al cálculo de los vértices de grado mayor, cuando se trata de obtener el mejor R_1 . En la práctica este método funciona sumamente rápido para gráficas pequeñas, en nuestros experimentos lo utilizamos en gráficas de hasta 10 vértices.

Ejemplo 1.5.2. Consideremos la gráfica $G = (V, E)$, con $V = \{1, 2, 3, \dots, 9\}$ y $E = \{\{1, 6\}, \{2, 6\}, \{2, 8\}, \{3, 5\}, \{3, 7\}, \{4, 9\}, \{5, 7\}, \{5, 9\}, \{7, 9\}\}$

La matriz de adyacencia es la siguiente:

$$M = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Iniciamos con el nodo raíz, las etiquetas en la parte superior, y la manera de mostrarlas, representan toda una clase de permutaciones. En el proceso se aprecia como se van descartando una gran cantidad de permutaciones en cada nivel, hasta obtener una única permutación y por lo tanto la forma canónica de la matriz de adyacencia de la gráfica.

Nodo 0,1

	{1, 2, 3, 4, 5, 6, 7, 8, 9}	Sumas
1	0 0 0 0 0 1 0 0 0	1
2	0 0 0 0 0 1 0 1 0	2
3	0 0 0 0 1 0 1 0 0	2
4	0 0 0 0 0 0 0 0 1	1
5	0 0 1 0 0 0 1 0 1	3
6	1 1 0 0 0 0 0 0 0	2
7	0 0 1 0 1 0 0 0 1	3
8	0 1 0 0 0 0 0 0 0	1
9	0 0 0 1 1 0 1 0 0	3

Sumando los elementos de cada renglón obtenemos que los primer vértices pueden ser $w = (5, 7, 9)$, así los nodos del primer nivel son:

- $Nodo_{1,1} = (\{5\}, N(5), N^c[5])$.
- $Nodo_{1,2} = (\{7\}, N(7), N^c[7])$.
- $Nodo_{1,3} = (\{9\}, N(9), N^c[9])$.

Nodo 1,1

	({5},	{3, 7, 9},	{1, 2, 4, 6, 8})	Sumas						
5	0	1	1	1	0	0	0	0	0	
3	1	0	1	0	0	0	0	0	0	1
7	1	1	0	1	0	0	0	0	0	2
9	1	0	1	0	0	0	1	0	0	1
1	0	0	0	0	0	0	0	1	0	
2	0	0	0	0	0	0	0	1	1	
4	0	0	0	1	0	0	0	0	0	
6	0	0	0	0	1	1	0	0	0	
8	0	0	0	0	0	1	0	0	0	

Nodo 1,2

	({7},	{3, 5, 9},	{1, 2, 4, 6, 8})	Sumas						
7	0	1	1	1	0	0	0	0	0	
3	1	0	1	0	0	0	0	0	0	1
5	1	1	0	1	0	0	0	0	0	2
9	1	0	1	0	0	0	1	0	0	1
1	0	0	0	0	0	0	0	1	0	
2	0	0	0	0	0	0	0	1	1	
4	0	0	0	1	0	0	0	0	0	
6	0	0	0	0	1	1	0	0	0	
8	0	0	0	0	0	1	0	0	0	

Nodo 1,3

	({9},	{4, 5, 7},	{1, 2, 3, 6, 8})	Sumas						
9	0	1	1	1	0	0	0	0	0	
4	1	0	0	0	0	0	0	0	0	0
5	1	0	0	1	0	0	1	0	0	1
7	1	0	1	0	0	0	1	0	0	1
1	0	0	0	0	0	0	0	1	0	
2	0	0	0	0	0	0	0	1	1	
3	0	0	1	1	0	0	0	0	0	
6	0	0	0	0	1	1	0	0	0	
8	0	0	0	0	0	1	0	0	0	

Capítulo 1

En este primer nivel siempre sucede que todos los nodos tienen el mismo R_1 , por lo que los tres nodos podrían ramificar y generar nueve nuevos nodos.

Sin embargo, es posible descartar ramificaciones examinando parte de la matriz de adyacencia de cada nodo. Incluso es posible saber la cantidad de los primeros unos que aparecerán en cada R_2 . Considere en cada nodo la submatriz determinada por los vértices $N(w_i)$ ³, al sumar los elementos de cada renglón en las submatrices se obtiene la cantidad de unos que aparecerán al principio en cada R_2 . Con este dato se pueden descartar las ramificaciones cuya suma que sea menor a la máxima suma.

De esta manera el $Nodo_{1,1}$ ramifica sólo tomando al índice 7 como segundo vértice y el $Nodo_{1,2}$ ramificaría con el vértice 5, pero no es necesario, la matriz de adyacencia es idéntica (hemos detectado un automorfismo). El $Nodo_{1,3}$ no ramifica, pues las tres posibles ramificaciones se descartan en la preselección. Hasta el momento ya se tiene que el primer vértice es 5 y el segundo es 7. El único nodo del nivel dos es:

Nodo 2,1

	({5}, {7}, {3, 9}, {1, 2, 4, 6, 8}) Sumas									
5	0	1	1	1	0	0	0	0	0	0
7	1	0	1	1	0	0	0	0	0	
3	1	1	0	0	0	0	0	0	0	0
9	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	0	0	1	0	
2	0	0	0	0	0	0	0	1	1	
4	0	0	0	1	0	0	0	0	0	
6	0	0	0	0	1	1	0	0	0	
8	0	0	0	0	0	1	0	0	0	

En este nodo tenemos que $N_1 = \{3, 9\}$ y $N_2 = \{1, 2, 4, 6, 8\}$. Las sumas de la submatriz indican que cualquiera de los dos 3 o 9 pueden ser el tercer vértice, aunque la entrada $9,4$ ya nos indica que pasará adelante. Así, los nodos del nivel tres son:

³En las matrices de adyacencia aparecen ya en un sólo bloque, esto se debe a la forma de particionar los índices.

Nodo 3,1

	({5}, {7}, {3}, {9}, {1, 2, 4, 6, 8})								
5	0	1	1	1	0	0	0	0	0
7	1	0	1	1	0	0	0	0	0
3	1	1	0	0	0	0	0	0	0
9	1	1	0	0	0	0	1	0	0
1	0	0	0	0	0	0	0	1	0
2	0	0	0	0	0	0	0	1	1
4	0	0	0	1	0	0	0	0	0
6	0	0	0	0	1	1	0	0	0
8	0	0	0	0	0	1	0	0	0

Nodo 3,2

	({5}, {7}, {9}, {3}, {4}, {1, 2, 6, 8})								
5	0	1	1	1	0	0	0	0	0
7	1	0	1	1	0	0	0	0	0
9	1	1	0	0	1	0	0	0	0
3	1	1	0	0	0	0	0	0	0
4	0	0	1	0	0	0	0	0	0
1	0	0	0	0	0	0	0	1	0
2	0	0	0	0	0	0	0	1	1
6	0	0	0	0	0	1	1	0	0
8	0	0	0	0	0	0	1	0	0

El nodo que ramifica es $Nodo_{3,2}$ pues tiene el mejor R_3 , en este nodo N_1 es unitario, así forzosamente ramifica con el vértice 3. El único nodo del nivel cuatro es:

Nodo 4,1

	({5}, {7}, {9}, {3}, {4}, {1, 2, 6, 8})								
5	0	1	1	1	0	0	0	0	0
7	1	0	1	1	0	0	0	0	0
9	1	1	0	0	1	0	0	0	0
3	1	1	0	0	0	0	0	0	0
4	0	0	1	0	0	0	0	0	0
1	0	0	0	0	0	0	0	1	0
2	0	0	0	0	0	0	0	1	1
6	0	0	0	0	0	1	1	0	0
8	0	0	0	0	0	0	1	0	0

En este $Nodo_{4,1}$ también N_1 es unitario, ramifica con el vértice 4 como quinto vértice.

Nodo 5,1

	({5}, {7}, {9}, {3}, {4}, {1, 2, 6, 8})									Sumas
5	0	1	1	1	0	0	0	0	0	
7	1	0	1	1	0	0	0	0	0	
9	1	1	0	0	1	0	0	0	0	
3	1	1	0	0	0	0	0	0	0	
4	0	0	1	0	0	0	0	0	0	
1	0	0	0	0	0	0	0	1	0	1
2	0	0	0	0	0	0	0	1	1	2
6	0	0	0	0	0	1	1	0	0	2
8	0	0	0	0	0	0	1	0	0	1

En principio el $Nodo_{5,1}$ podría ramificar con los vértices 1, 2, 6 y 8, sin embargo la preselección descarta a los vértices 1 y 8. El nivel seis tiene dos nodos.

	({5}, {7}, {9}, {3}, {4}, {2}, {6, 8}, {1})									Sumas
5	0	1	1	1	0	0	0	0	0	
7	1	0	1	1	0	0	0	0	0	
9	1	1	0	0	1	0	0	0	0	
3	1	1	0	0	0	0	0	0	0	
4	0	0	1	0	0	0	0	0	0	
2	0	0	0	0	0	0	1	1	0	
6	0	0	0	0	0	1	0	0	1	0
8	0	0	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	

	({5}, {7}, {9}, {3}, {4}, {6}, {1, 2}, {8})									Sumas
5	0	1	1	1	0	0	0	0	0	
7	1	0	1	1	0	0	0	0	0	
9	1	1	0	0	1	0	0	0	0	
3	1	1	0	0	0	0	0	0	0	
4	0	0	1	0	0	0	0	0	0	
6	0	0	0	0	0	0	1	1	0	
1	0	0	0	0	0	1	0	0	0	0
2	0	0	0	0	0	1	0	0	1	0
8	0	0	0	0	0	0	0	1	0	

La preselección no descarta ramificaciones, aunque podría hacerlo si se amplía la comparación. El $Nodo_{6,1}$ ramifica con 2 y 6, el $Nodo_{6,2}$ ramifica con 1 y 2. El nivel siete tiene cuatro nodos.

Nodo 7,1

	({5}, {7}, {9}, {3}, {4}, {2}, {6}, {8}, {1})								
5	0	1	1	1	0	0	0	0	0
7	1	0	1	1	0	0	0	0	0
9	1	1	0	0	1	0	0	0	0
3	1	1	0	0	0	0	0	0	0
4	0	0	1	0	0	0	0	0	0
2	0	0	0	0	0	0	1	1	0
6	0	0	0	0	0	1	0	0	1
8	0	0	0	0	0	1	0	0	0
1	0	0	0	0	0	0	1	0	0

Nodo 7,2

	({5}, {7}, {9}, {3}, {4}, {2}, {8}, {6}, {1})								
5	0	1	1	1	0	0	0	0	0
7	1	0	1	1	0	0	0	0	0
9	1	1	0	0	1	0	0	0	0
3	1	1	0	0	0	0	0	0	0
4	0	0	1	0	0	0	0	0	0
2	0	0	0	0	0	0	1	1	0
8	0	0	0	0	0	1	0	0	0
6	0	0	0	0	0	1	0	0	1
1	0	0	0	0	0	0	0	1	0

Nodo 7,3

	({5}, {7}, {9}, {3}, {4}, {6}, {1}, {2}, {8})								
5	0	1	1	1	0	0	0	0	0
7	1	0	1	1	0	0	0	0	0
9	1	1	0	0	1	0	0	0	0
3	1	1	0	0	0	0	0	0	0
4	0	0	1	0	0	0	0	0	0
6	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	1	0	0	0
2	0	0	0	0	0	1	0	0	1
8	0	0	0	0	0	0	0	1	0

Nodo 7,4

	({5}, {7}, {9}, {3}, {4}, {6}, {2}, {1}, {8})								
5	0	1	1	1	0	0	0	0	0
7	1	0	1	1	0	0	0	0	0
9	1	1	0	0	1	0	0	0	0
3	1	1	0	0	0	0	0	0	0
4	0	0	1	0	0	0	0	0	0
6	0	0	0	0	0	0	1	1	0
2	0	0	0	0	0	1	0	0	1
1	0	0	0	0	0	1	0	0	0
8	0	0	0	0	0	0	1	0	0

Los nodos $Nodo_{7,2}$ y $Nodo_{7,3}$ se descartan pues no tienen el mejor R_7 y el nodo $Nodo_{7,4}$ se descarta por automorfismo.

Así hemos obtenido un único nodo con una partición fija (todos los subconjuntos son unitarios) y por lo tanto también se ha obtenido la forma canónica de la matriz de adyacencia de la gráfica G .

En la siguiente página se muestra el árbol que fué construido en este proceso.

Capítulo 1



Árbol generado en el proceso de etiquetado canónico, los nodos no enmarcados fueron descartados. Nótese que sólo fueron consideradas 11 representantes de clases de permutaciones, evitando trabajar con las $366880 = 9!$ posibles permutaciones.

Haber calculado la forma canónica de la gráfica anterior equivale a haber retiquetado los vértices de tal manera que ahora las aristas originales se han transformado de:

$$E = \{\{1, 6\}, \{2, 6\}, \{2, 8\}, \{3, 5\}, \{3, 7\}, \{4, 9\}, \{5, 7\}, \{5, 9\}, \{7, 9\}\}$$

a

$$\pi(E) = \{\{5, 2\}, \{7, 2\}, \{7, 1\}, \{9, 4\}, \{9, 6\}, \{3, 1\}, \{4, 6\}, \{4, 1\}, \{6, 1\}\},$$

donde

$$\pi = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 5 & 7 & 9 & 3 & 4 & 2 & 6 & 8 & 1 \end{pmatrix}.$$

1.6. Complejos simpliciales

En cierto sentido el concepto de complejo simplicial viene a generalizar al concepto de gráfica simple.

Los complejos simpliciales son utilizados en análisis topológicos de datos (ATD), que dicho sea de paso ha estado cobrando mucha importancia en los años recientes gracias a su utilidad.

Definición 1.6.1. Un **complejo simplicial** abstracto finito es un conjunto finito A junto con una colección Δ de subconjuntos de A que cumple la siguiente condición: Si $\sigma \in \Delta$ y $\tau \subset \sigma$, entonces $\tau \in \Delta$.

A cada elemento de Δ se le llama un *simplejo*, en particular a los conjuntos unitarios subconjuntos de A se les llama vértices de Δ y abusando un poco de la notación representaremos por v_i o por $\{v_i\}$.

Ejemplo 1.6.2. Sea $A = \{1, 2, 3, 4, 5, 6\}$, la siguiente familia de subconjuntos forma una estructura simplicial.

$$\begin{aligned} \Delta = \{ & \{\}, \{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{6\}, \\ & \{1, 2\}, \{2, 3\}, \{2, 4\}, \{3, 4\}, \{3, 5\}, \{3, 6\}, \{4, 5\}, \{4, 6\}, \{5, 6\}, \\ & \{2, 3, 4\}, \{3, 4, 5\}, \{4, 5, 6\}, \{3, 4, 6\}, \{3, 5, 6\}, \\ & \{3, 4, 5, 6\} \end{aligned}$$

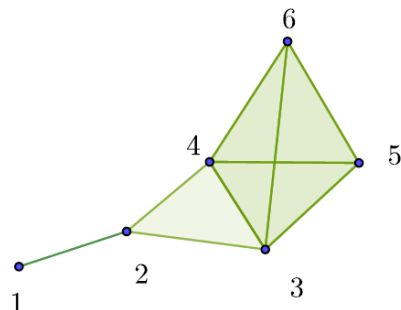
Se puede verificar que todo subconjunto de $\tau \in \Delta$ también es elemento de Δ .

Según sea la cardinalidad de cada simplejo se le llama: vértice, arista, triángulo, tetraedro, ..., etc. La dimensión del simplejo $\sigma \neq \{\}$ es $|\sigma| - 1$.

Definición 1.6.3. Un complejo simplicial concreto es la representación geométrica del complejo simplicial abstracto.

En la representación geométrica de cada simplejo se considera el espacio que encierra, por ejemplo para la terna $\{2, 3, 4\}$ la representación geométrica de incluye los puntos $\{2\}, \{3\}, \{4\}$, las aristas $\{2, 3\}, \{2, 4\}, \{3, 4\}$ y el interior delimitado por las dichas aristas. Algo similar sucede con el tetraedro $\{3, 4, 5, 6\}$.

De esta manera el siguiente complejo simplicial concreto correspondiente al complejo simplicial abstracto anterior (Ejemplo 1.6.2) es:



Observese que:

- $\{1\} \subset \{1, 2\}$.
- $\{2, 3\} \subset \{2, 3, 4\}$.
- $\{3, 4, 5\} \subset \{3, 4, 5, 6\}$.

y que los primeros conjuntos sólo están contenidos en un único conjunto, y que el segundo conjunto no está contenido en algún otro, (es maximal). Esto da pie a la siguiente definición.

Definición 1.6.4. Sean $\sigma, \tau \in \Delta$ con $\dim(\sigma) = \dim(\tau) + 1$. Si $\tau \subset \sigma$, τ no está contenido en otro simplejo y σ no está contenido en ningún simplejo, entonces a (σ, τ) se le llama **pareja libre** del complejo simplicial Δ y a τ se le llama **cara libre**.

A σ se le llama supercara de τ y a τ cara de σ .

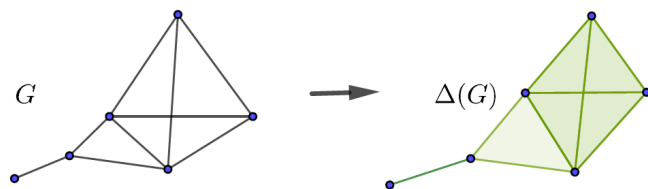
En el ejemplo anterior, los siguientes son algunos parejas libres de Δ ($\{1, 2\}, \{1\}$), ($\{2, 3, 4\}, \{2, 3\}$), ($\{3, 4, 5, 6\}, \{3, 5, 6\}$), ($\{3, 4, 5, 6\}, \{3, 5, 6\}$).

Nótese que $(\{2, 3, 4\}, \{3, 4\})$ no es pareja libre de Δ .

Dada una gráfica simple es posible construir el complejo simplicial $\Delta(G)$ de la siguiente manera:

Consideremos el conjunto de vértices $V = \{v_1, v_2, \dots, v_n\}$ y al conjunto de aristas E .

- Para empezar $\{v_i\} \in \Delta(G)$, además $\{v_i, v_j\} \in \Delta(G)$ para $i, j = 1, 2, \dots, n$. Toda arista de E , también pertenece a $\Delta(G)$
- Para obtener el resto de los simplejos será necesario identificar todas las subgráficas completas de G para obtener el total de los simplejos de $\Delta(G)$, la máxima dimensión del simplejo está acotada por el mayor grado de los vértices.



Al complejo obtenido se le llama *complejo simplicial de Whitney*, también se le puede llamar complejo de completas o **flag complex** entre otros. A este procedimiento de construir $\Delta(G)$ le llamaremos método de expansión.

Capítulo 2

Familias de gráficas

Existe una amplia variedad de familias de gráficas, incluso si nos restringimos a gráficas simples y con un número finito de vértices. No es difícil encontrar colecciones con cientos, miles o incluso millones de gráficas, donde el lector puede acceder a dichos bancos para apreciar representaciones geométricas, para tratar descubrir propiedades de las gráficas o útiles para validar o refutar conjeturas sobre propiedades de las gráficas.

2.1. Familia de gráficas conexas

Es posible diseñar diferentes procedimientos para generar colecciones completas de gráficas finitas conexas con un número acotado de vértices. Un procedimiento poco recomendable consiste en generar todas las posibles matrices de adyacencia con $|V|$ vértices y $|E|$ aristas, lo primero a verificar es que la gráfica generada sea conexa, y aunque existen métodos eficientes para verificar la conexidad, es una tarea que se puede evitar. Una vez que se ha verificado la conexidad de la gráfica, es necesario verificar que no sea isomorfa a alguna de las gráficas ya agregadas a la colección para ser agregada a la misma.

En una búsqueda somera no hemos encontrado explicaciones a detalle de cómo proceden a generar estas colecciones de gráficas, sin embargo es recomendable ofrecer detalles a fin de poder proponer una manera estándar de generar y representar gráficas.

El procedimiento descrito a continuación evita la verificación de la conexidad pues contruye gráficas a través de agregar elementos a una gráfica conexa de tal manera que la nueva gráfica es también conexa.

Para generar la colección de las gráficas conexas, comenzamos con la gráfica conexa $K(1)$, realizando las siguientes operaciones para obtener otras gráficas conexas:

Op1: Agregar el vértice v_{n+1} y la arista $\{v_i, v_{n+1}\}$ para algún i de $\{1, 2, \dots, n\}$.

Op2: Agregar arista $\{v_i, v_j\}$ a G si $\{v_i, v_j\} \notin E$ con $i \neq j$.

Observaciones. 2.1.1.

- La operación **Op1** se lleva a cabo en la matriz de adyacencia agregando al final una columna de ceros (y el correspondiente renglón de ceros) y sustituir a alguna de estas entradas por un uno (excepto en la diagonal). Si la matriz original es de orden k , esta operación tiene k opciones.

- La operación **Op2** consiste en sustituir en la matriz de adyacencia un cero fuera de la diagonal por un uno.
- En ambas operaciones, dada la simetría de la matriz es necesario agregar el correspondiente uno del otro lado de la diagonal principal.

Cada gráfica generada se usa para generar nuevas gráficas conexas. Aquí también, para cada gráfica candidato generada, se necesita verificar si es o no isomorfa a alguna ya agregada a la colección. Al igual que en varios trabajos donde se generan bancos de gráficas, tomamos como representante de la clase de equivalencia bajo isomorfismo a la primera gráfica generada de cada clase.

Esto significa que, en los procesos de elaboración de colecciones, no suele realizarse trabajo adicional buscando mejores representantes de la clase de equivalencia.

En el procedimiento que se desarrolla a continuación, se organiza la colección en subconjuntos de gráficas con el par $[|V|, |E|]$, así el primer subconjunto es $[1, 0]$ y contiene únicamente a $K(1)$.

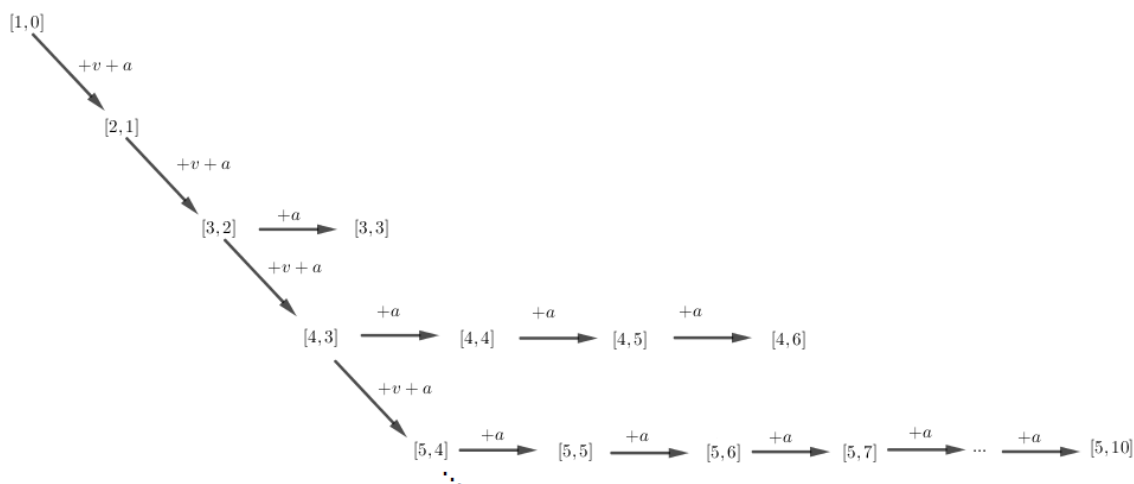
Existen varias formas de organizar la construcción de estos subconjuntos. Por ejemplo, suponga que se desea generar todas las gráficas conexas a lo sumo de n vértices.

Una manera de generar la colección puede empezar por generar todos los árboles empezando con $[1, 0]$, siguiendo con $[2, 1]$, luego $[3, 2]$, así hasta llegar a $[n, n - 1]$, sólo aplicando **Op1**.

Esto es, a cada gráfica del subconjunto de árboles $[k, k - 1]$ se le agrega un nuevo vértice pegándolo con una arista a un vértice pre-existente. Al igual que en otros procedimientos, para agregar la nueva gráfica, es necesario verificar que no sea isomorfa a alguna gráfica ya obtenida.

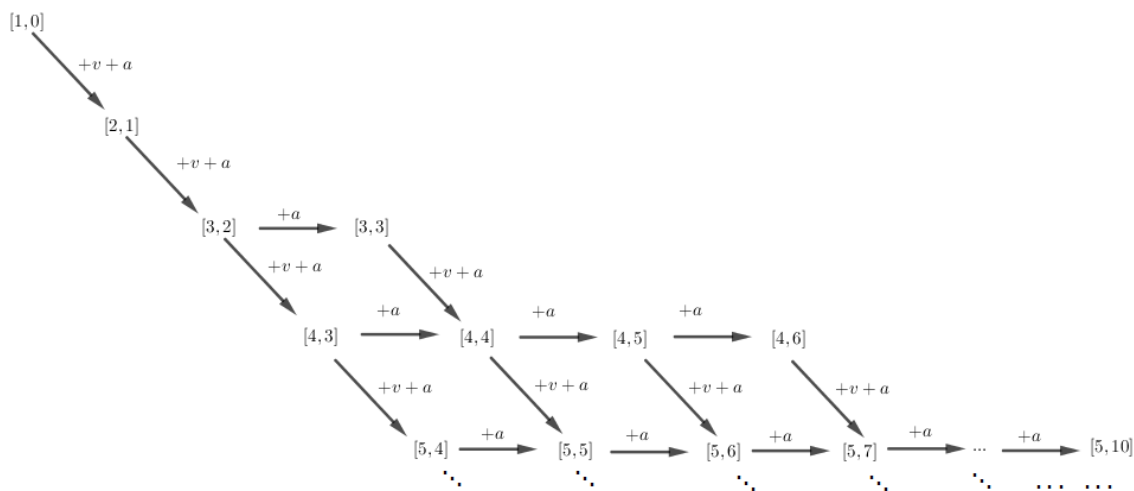
Una vez obtenidos todos los árboles, se procede a pegar aristas utilizando sólo la operación **Op2**. Esto es, del subconjunto de árboles de k vértices $[k, k - 1]$, generamos todas las gráficas no isomorfas del subconjunto de graficas $[k, k]$ y luego las del subconjunto $[k, k + 1]$, hasta llegar al subconjunto de gráficas completas $[k, k(k - 1)/2]$.

En la siguiente figura cada nodo representa a un subconjunto de gráficas.



En este procedimiento no se ha mencionado el orden en el cual se modifica a los elementos de la matriz de adyacencia para obtener gráficas candidato.

Enseguida se muestra el esquema para generar una colección de gráficas conexas, muy similar al anterior, pero con resultados que juzgamos más apropiados para efectos de visualización de gráficas. De hecho lo proponemos como un estándar para la generación de gráficas conexas. En lugar de darle una estructura de árbol al mapa de subconjuntos de gráficas con nodos $[|V|, |E|]$, le damos la siguiente estructura de malla:



Para generar los subconjuntos $[|V|, |E|]$ le damos prioridad a la operación **Op1** sobre la operación **Op2**, esto es, siempre que sea posible ¹ tomamos gráficas del subconjunto $[|V| - 1, |E| - 1]$ y agregamos vértice pegado con una arista a un vértice pre-existente, una vez agotadas todas las posibilidades se procede a tomar gráficas del subconjunto $[|V|, |E| - 1]$ y le agregamos un arista, obviamente donde no existe. Esto último no es posible cuando $[|V|, |E|]$ son árboles.

Es necesario mencionar que cuando se realiza la operación **Op1**, el pegado de vértices, empieza a generar gráficas candidato pegando primero un vértice nuevo con una arista a vértices de índice mayor. En seguida se muestra el algoritmo 1:

¹Por ejemplo esta operación no es posible para generar los subconjuntos $[3, 3]$, ni $[4, 5]$, tampoco $[4, 6]$ entre muchos otros.

$$\begin{bmatrix} 0 & a_{1,2} & a_{1,3} & \dots & a_{1,n-1} & a_{1,n} \\ a_{1,2} & 0 & a_{2,3} & \dots & a_{2,n-1} & a_{2,n} \\ a_{1,3} & a_{2,3} & 0 & \dots & a_{3,n-1} & a_{3,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{1,n-1} & a_{2,n-1} & a_{3,n-1} & \dots & 0 & a_{n-1,n} \\ a_{1,n} & a_{2,n} & a_{3,n} & \dots & a_{n-1,n} & 0 \end{bmatrix} \longrightarrow \begin{bmatrix} 0 & a_{1,2} & a_{1,3} & \dots & a_{1,n-1} & a_{1,n} & 0 \\ a_{1,2} & 0 & a_{2,3} & \dots & a_{2,n-1} & a_{2,n} & 0 \\ a_{1,3} & a_{2,3} & 0 & \dots & a_{3,n-1} & a_{3,n} & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ a_{1,n-1} & a_{2,n-1} & a_{3,n-1} & \dots & 0 & a_{n-1,n} & 0 \\ a_{1,n} & a_{2,n} & a_{3,n} & \dots & a_{n-1,n} & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 \end{bmatrix}$$

Algorithm 1: add.vertex.edge.to.G

Input : Una matriz de adyacencia M de una gráfica G conexa con $n = |V|$ y $|E|$ aristas.

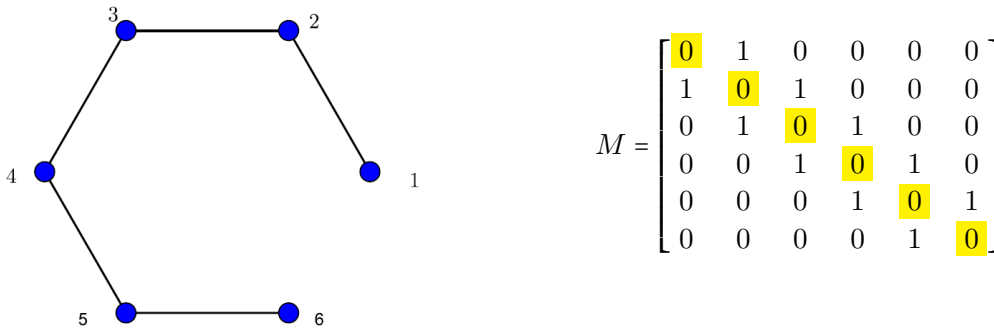
Output: Un conjunto de matrices de adyacencia de gráficas conexas de $n + 1$ vértices y $|E| + 1$ aristas.

```

1 for  $i \leftarrow 1$  to  $n + 1$  do
2   |  $a_{i,n+1} \leftarrow 0$ 
3   |  $a_{n+1,i} \leftarrow 0$ 
4 end
5  $k \leftarrow 0$ 
6 for  $i \leftarrow n$  to 1 do
7   |  $a_{i+1,n+1} \leftarrow 0$ 
8   |  $a_{n+1,i+1} \leftarrow 0$ 
9   |  $a_{i,n+1} \leftarrow 1$ 
10  |  $a_{n+1,i} \leftarrow 1$ 
11  |  $k \leftarrow k + 1$ 
12  |  $M_k \leftarrow M$ 
13 end

```

El efecto logrado en esta estrategia se refleja en las gráficas camino, pues unen a los vértices en orden, de tal manera que visualmente se identifican muy fácil sus representaciones geométricas, incluso su representación matricial. Un ejemplo se muestra en la siguiente figura:



Estas gráficas surgen inmediatamente después de lograr una gráfica completa, es decir

es la primera gráfica con tal cantidad de vértices, así el resto de las gráficas que se derivan, se aprecian menos complicadas.

Para la operación **Op2**, pegado de aristas, se realiza utilizando el siguiente algoritmo:

Algorithm 2: `add.edge.to.G`

Input : Una matriz de adyacencia M de una gráfica G conexa con $n = |V|$ y $|E|$ aristas.

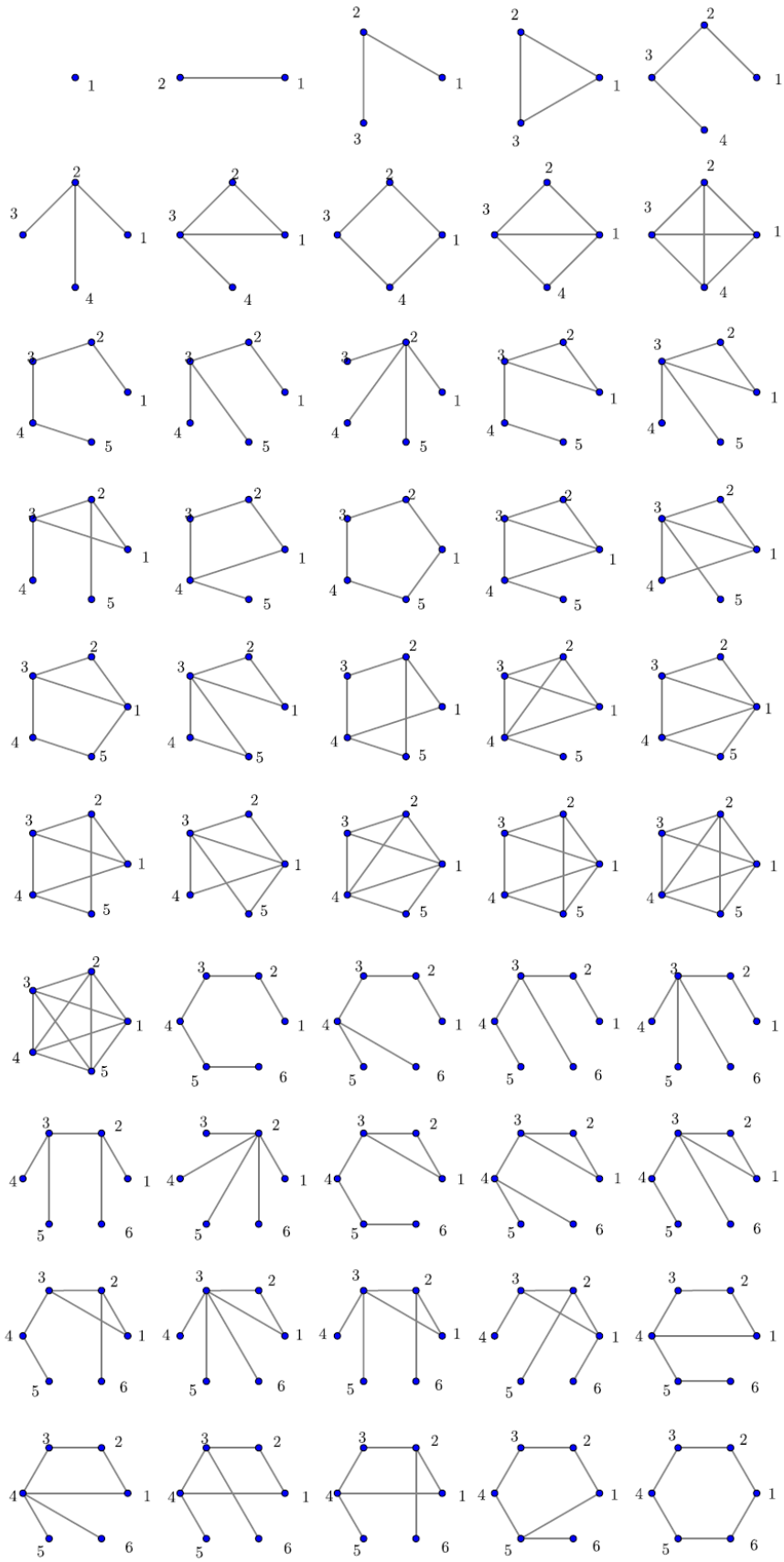
Output: Un conjunto de matrices de adyacencia de gráficas conexas de n vértices y $|E| + 1$ aristas.

```

1  $new \leftarrow 0$ 
2 for  $i \leftarrow 1$  to  $n - 1$  do
3   for  $j \leftarrow 1$  to  $n$  do
4     if  $a_{i,j} = 0$  then
5        $new \leftarrow new + 1$ 
6        $a_{i,j} \leftarrow 1$ 
7        $a_{j,i} \leftarrow 1$ 
8        $M_{new} \leftarrow M$ 
9        $a_{i,j} \leftarrow 0$ 
10       $a_{j,i} \leftarrow 0$ 
11     end
12   end
13 end

```

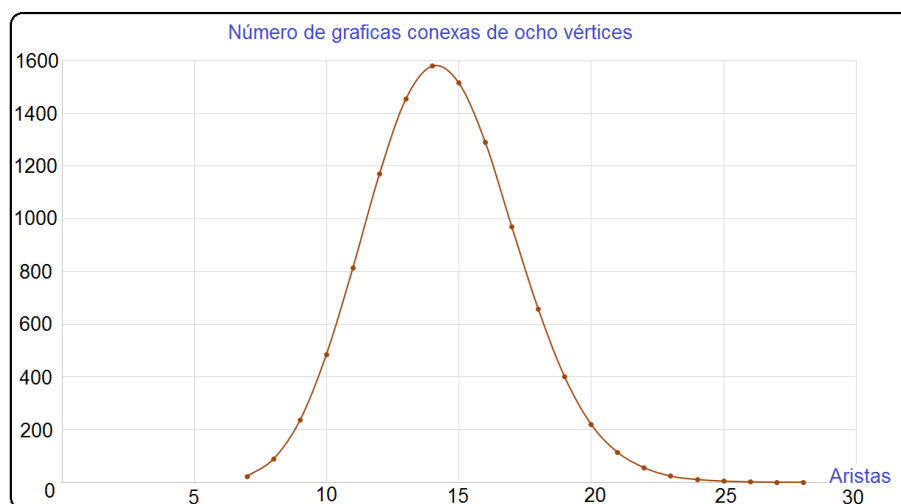
Esto significa que empezamos a pegar aristas entre los primeros pares de vértices “libres”, empezando con el primer renglón recorriéndolo de izquierda a derecha y concluyendo la revisión con el renglón $n-1$, para cada cero detectado se sustituye por un uno para generar una gráfica candidato. Antes de revisar otra entrada de la matriz, la matriz modificada vuelve a su estado original, de tal manera que todas las graficas candidato tienen sólo una arista más, respecto a la gráfica de donde fueron obtenidas. Enseguida se muestra una figura con las primeras gráficas generadas con este procedimiento.



Si alguna gráfica pequeña no parece en la lista, muy probablemente es isomorfa a alguna ya mostrada, o no fue de las primeras 50 gráficas generadas. Lo destacable de este procedimiento es que se obtienen graficas visualmente comprensibles, sin necesidad de realizar la tarea especial de búsqueda de mejores representantes de cada clase de isomorfismo.

En el Apéndice A.1 se presenta una revisión del proceso, donde se especifica cuando una gráfica candidato es isomorfa a una gráfica ya obtenida.

En la siguiente figura se muestra la distribución del número de gráficas de 8 vértices respecto a la cantidad de aristas.



Se puede apreciar que aproximadamente en el tercio del dominio de $[n - 1, \frac{n(n-1)}{2}]$, es decir, aproximadamente en entero $\left(\frac{(n-1)(n-4)}{6}\right)$ aristas, se alcanza el máximo número de gráficas de $n = |V|$ vértices.

2.1.1. Aspectos técnicos

La cantidad de gráficas conexas crece rápidamente respecto a la cantidad de vértices. Una limitante para el número máximo de vértices a considerar es la capacidad de almacenamiento secundario para las gráficas generadas.

Para el cálculo de gráficas se utilizó una computadora personal de escritorio procesador Intel i8, de 8 núcleos, 8 Gb de memoria RAM, con sistema operativo Windows 8, los códigos fueron desarrollados con el compilador de Dev C/C++.

Al trabajar con gráficas de diez vértices comenzamos a requerir más memoria RAM o memoria secundaria para el almacenamiento temporal de las gráficas obtenidas. Los códigos no implementan estructuras especializadas de datos. Por estos motivos, el trabajo está limitado a cantidades pequeñas de vértices.

De hecho, se pueden realizar mejoras a los códigos y las estructuras de datos utilizadas, para aumentar en el número de vértices. Sin embargo, no se pretendría llegar muy lejos pues la cantidad de gráficas aumenta rápidamente a medida que el número de vértices aumenta, ver [19].

En el repositorio [3] se comparten los códigos escritos en C/C++, los archivos contienen matrices de adyacencia de las gráficas generadas así como algunas representaciones geométricas.

Desempeño. Para tener un idea de las bondades del etiquetamento canónico, al resolver el problema de isomorfismo, utilizado en el proceso de generar la colección de gráficas conexas hasta ocho vértices, mencionamos los siguientes datos:

- Utilizando una implementación simple de los códigos compartidos en [9] para detectar isomorfismos entre cada par de gráficas, requirió aproximadamente una semana de cálculos.
- Resolviendo en cada paso el problema de isomorfismo en base a la utilización de permutaciones, requirió aproximadamente 2.5 horas.
- Utilizando etiquetamento canónico y identificando a cada gráfica con su número máximo, necesita 1.5 segundos aproximadamente. Para generar la colección de gráficas conexas hasta nueve vértices, el tiempo sube a tres minutos aproximadamente.

2.2. Familia de gráficas \mathfrak{J} -contraíbles

En [15] fue definida la siguiente familia, y sus elementos son llamadas gráficas contraíbles, para no confundir con el concepto topológico de contraíble, a dicha familia le llamaremos \mathfrak{J} -contraíble.

Definición 2.2.1. Sea \mathfrak{J} la familia de gráficas definidas por

1. La gráfica trivial $K(1)$ está en \mathfrak{J} .
2. Cualquier gráfica de \mathfrak{J} puede ser obtenida de $K(1)$ por las siguientes transformaciones.
 - (I1) Borrado de vértice. Un vértice v de una gráfica G puede ser eliminado si $N_G(v) \in \mathfrak{J}$. En cuyo caso el vértice es llamado vértice \mathfrak{J} -contraíble.
 - (I2) Pegado de un vértice. Si una subgráfica G_1 de la gráfica G está en \mathfrak{J} , entonces el vértice v puede ser pegado a la gráfica G de tal manera que $N_G(v) = G_1$.
 - (I3) Borrado de arista. La arista $\{v_1, v_2\}$ de una gráfica G puede ser borrada si $N_G(v_1, v_2) \in \mathfrak{J}$. En cuyo caso la arista es llamada arista \mathfrak{J} -contraíble.
 - (I4) Pegado de arista. Sean dos vértices v_1 y v_2 de una gráfica G no adyacentes. La arista $\{v_1, v_2\}$ puede ser pegada si $N_G(v_1, v_2) \in \mathfrak{J}$.

Si G pertenece a \mathfrak{J} , entonces G es llamada **gráfica \mathfrak{J} -contraíble**.

A las transformaciones (I1)-(I4) en [16] le llamaremos transformaciones \mathfrak{J} -contraíbles. Las transformaciones \mathfrak{J} -contraíbles son usadas en espacios moleculares, ver [16], para más explicaciones. Además, en [15] se probó que las transformaciones no cambian al grupo de homología de una gráfica, para cualquier grupo conmutativo de coeficientes A , así los elementos de \mathfrak{J} tienen grupos triviales de A -homología.

En [8, Lema 3.4] se muestra que el borrado de arista (pegado) puede realizarse con la composición de pegado (borrado) de un vértice y borrado (pegado) de un vértice.

Esto implica que la familia \mathfrak{J} – *contraíble* puede obtenerse sólo con las operaciones (I1) y (I2). Por otra parte es posible mostrar, con base en cualquiera de las dos gráficas que se mencionan en el primer párrafo de la Sección 2.6, que \mathfrak{J} posee una subfamilia propia si nos restringimos a las transformaciones (I2) y (I4).

Definición 2.2.2. Sea \mathfrak{J}_f la familia de gráficas definidas por

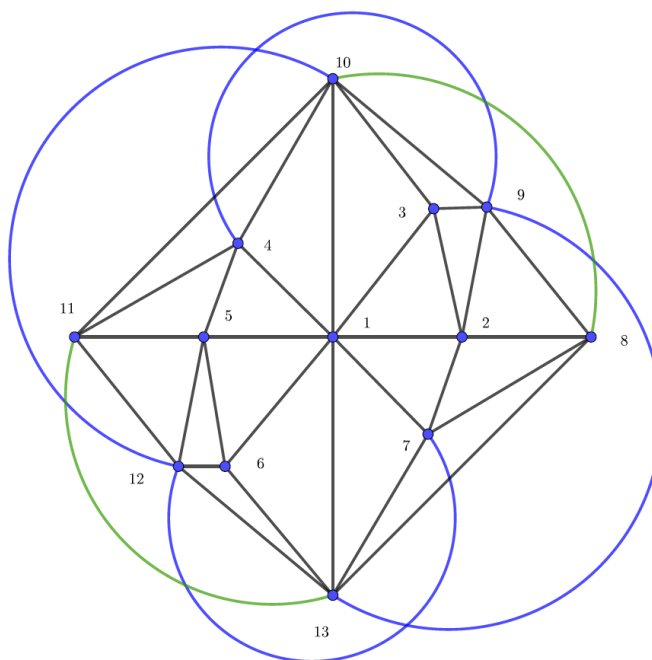
1. La gráfica trivial $K(1)$ está en \mathfrak{J}_f .
2. Cualquier gráfica de \mathfrak{J}_f puede ser obtenida de $K(1)$ por las siguientes transformaciones.
 - (I2) Pegado de un vértice. Si una subgráfica G_1 de la gráfica G está en \mathfrak{J}_f , entonces el vértice v puede ser pegado a la gráfica G de tal manera que $N_G(v) = G_1$.
 - (I4) Pegado de arista. Sean dos vértices v_1 y v_2 de una gráfica G no adyacentes. La arista $\{v_1, v_2\}$ puede ser pegada si $N_G(v_1, v_2) \in \mathfrak{J}_f$.

Si G pertenece a \mathfrak{J}_f , entonces G es llamada **gráfica fuertemente \mathfrak{J} -contraíble**.

Observación. 2.2.3. En [16, Teo. 3.8] Ivashchenko establece que la familia \mathfrak{J} puede ser construida a partir de $K(1)$, solamente por la iteración de la transformación (I2).

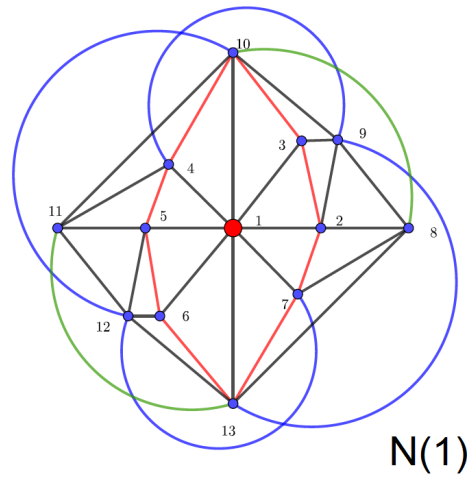
Dado que la operación de agregar vértice es reversible, el teorema mencionado, implicaría que toda gráfica de la familia \mathfrak{J}_f puede ser transformada en $K(1)$ solamente eliminando vértices en algún orden, lo cual lamentablemente no sucede.

La gráfica que se muestra a continuación es un ejemplo que comprueba la afirmación anterior.

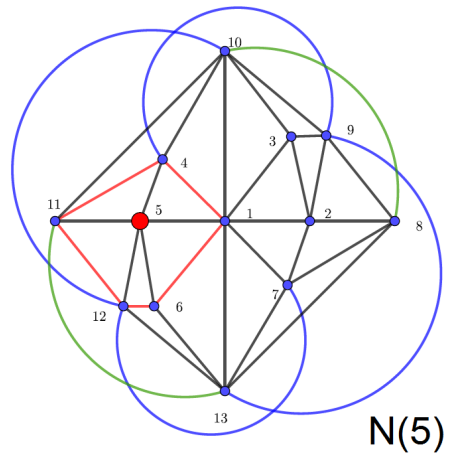
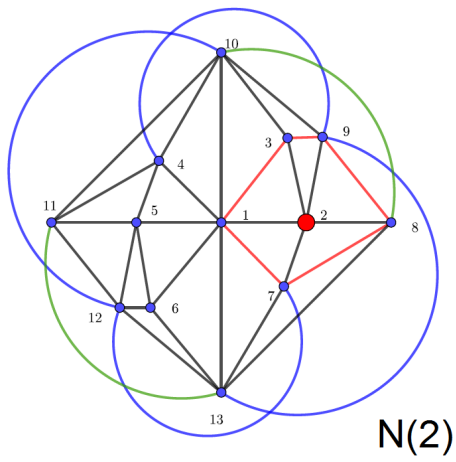


Se puede verificar que ningún vértice puede ser eliminado sólo con la operación de borrado de vértice, pues sus vecindades no son elementos de \mathfrak{J}_f .

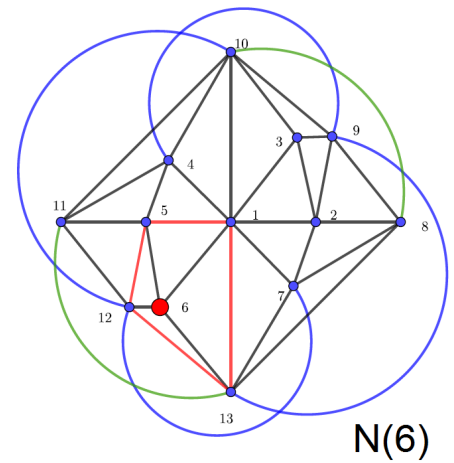
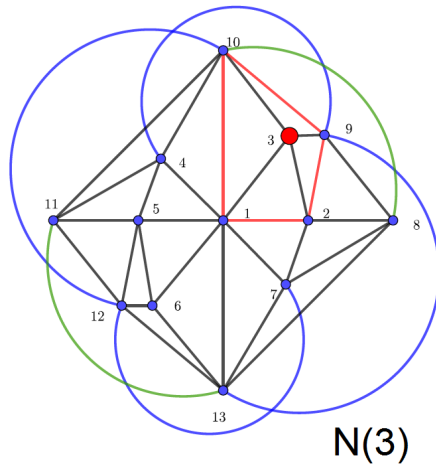
- $N(1)$ es un octágono (líneas rojas).



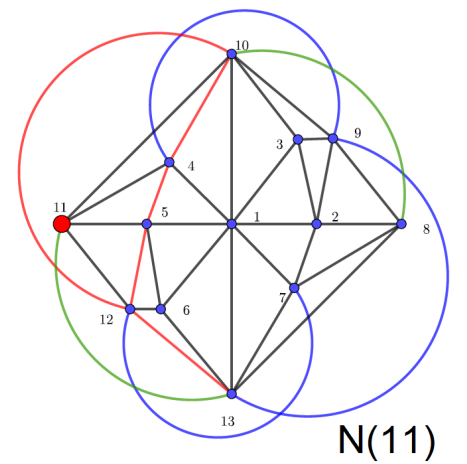
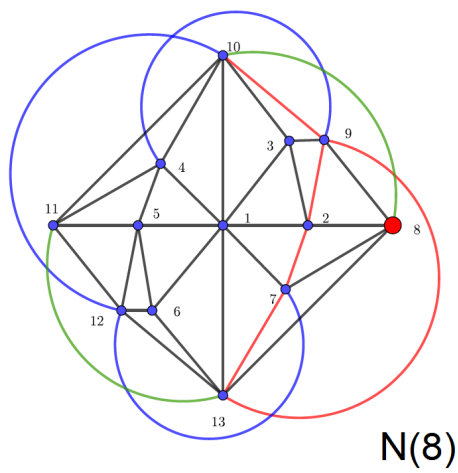
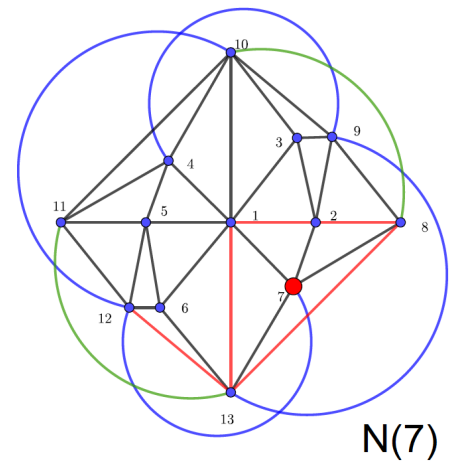
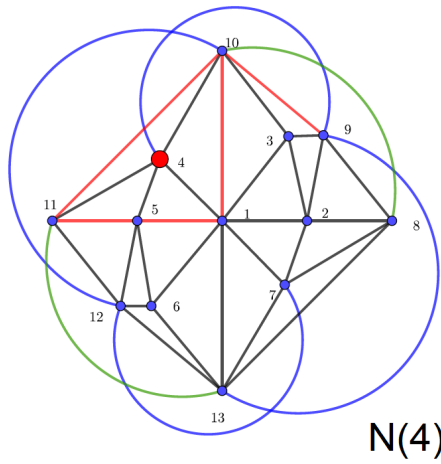
- $N(2)$ y $N(5)$ son pentágonos.



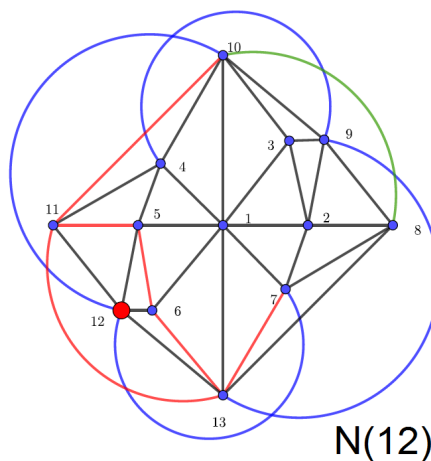
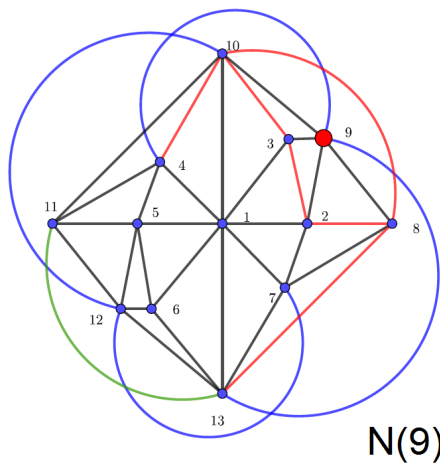
- $N(3)$ y $N(6)$ son cuadriláteros.



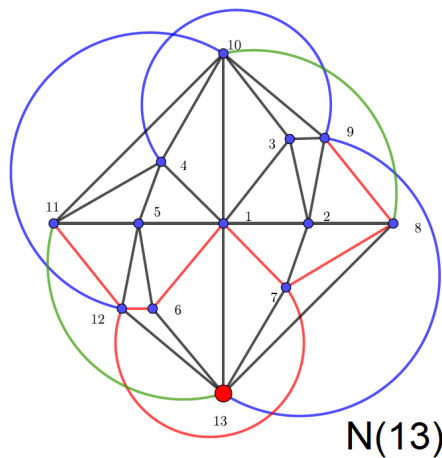
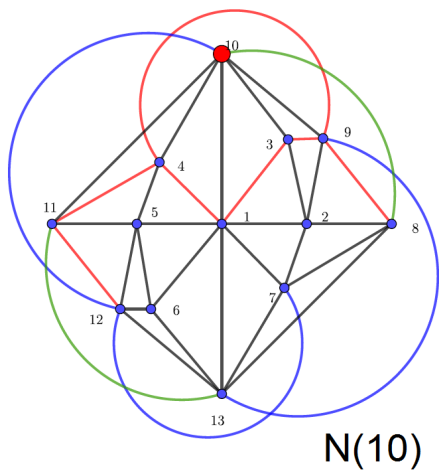
- N(4), N(7), N(8) y N(11) son cuadriláteros con una hoja.



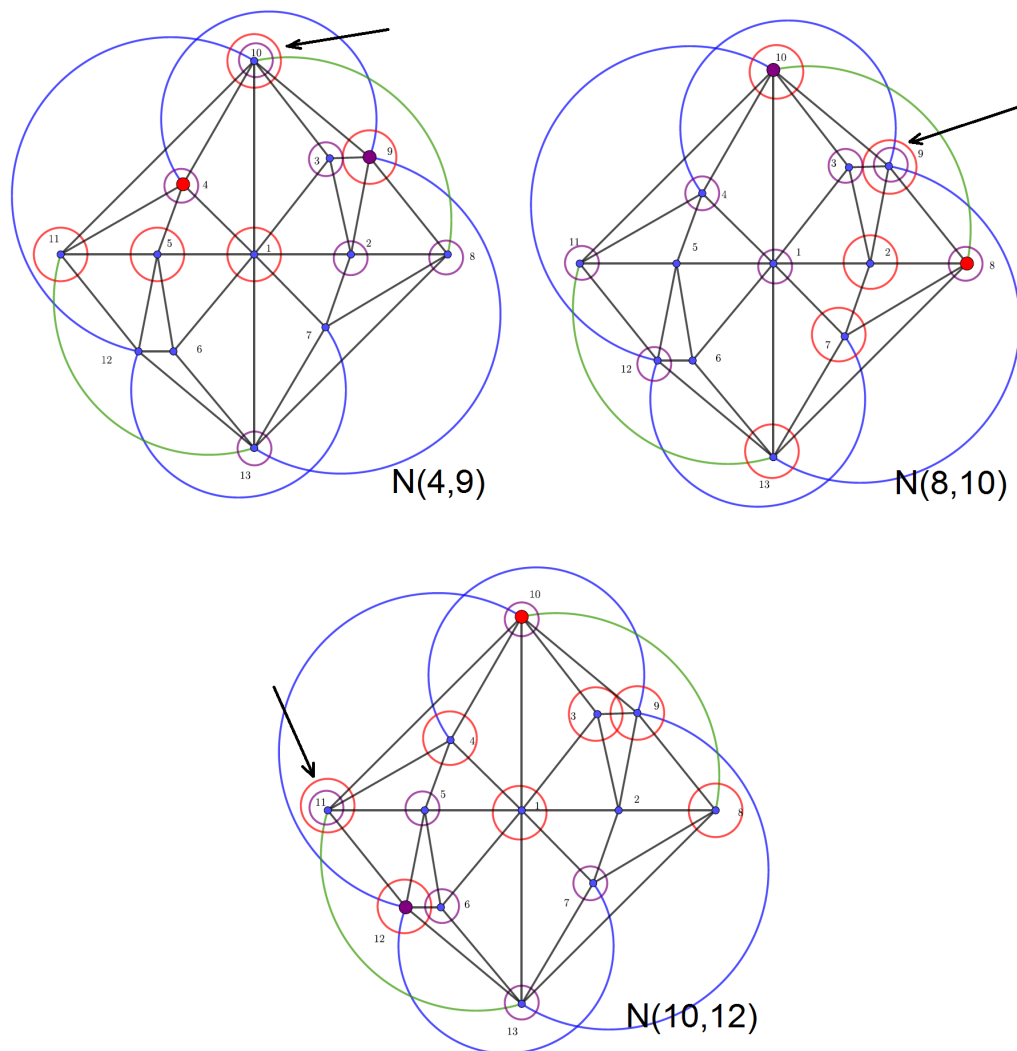
- N(9) y N(12) son cuadriláteros con dos hojas.



- N(10) y N(13) son un cuadriláteros con una hoja y con una cadena de logitud dos.



Sin embargo las aristas coloreadas (las aristas curvas) pueden ser eliminadas, ya que su vecindad abierta es un punto. Por simetría basta exhibir que tres aristas esencialmente distintas. Las aristas pueden ser borradas pues su vecindad es $K(1)$, como se muestra en las tres siguientes figuras:

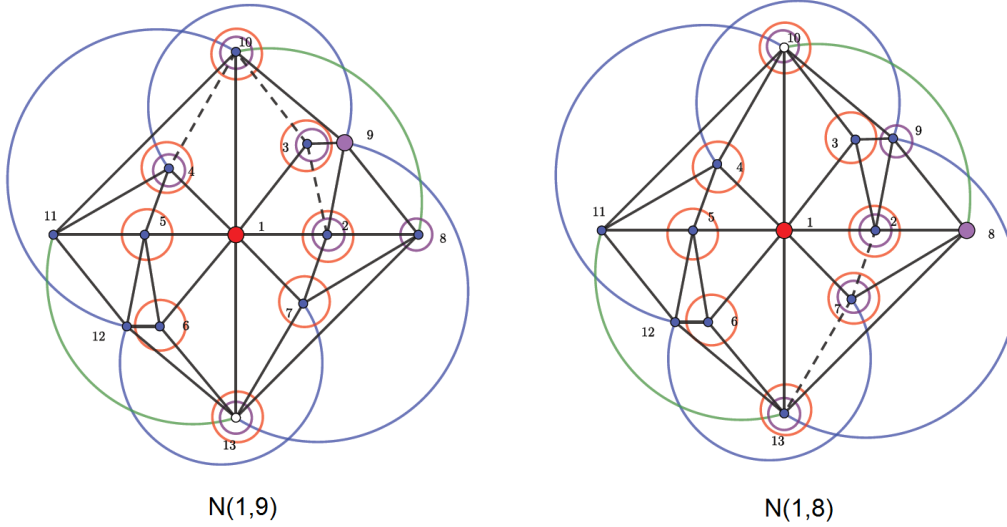


Esta gráfica se puede transformar a un punto, borrando las aristas coloreadas se obtiene la triangulación plana del disco, a la que se lleva al punto eliminando vértices.

Adicionalmente la gráfica anterior contradice al menos dos afirmaciones dadas en [16], a saber:

- Axioma 3.4: Supongamos que $G \in \mathcal{J}$ -contraíble y v un vértice que no está conectado con algunos vértices de G , entonces existe un vértice no adyacente $u \in G$, tal que $N_G(u, v)$ es \mathcal{J} -contraíble
- Teorema 3.5: Sea $G \in \mathcal{J}$ -contraíble con $|V| > 1$ entonces tiene al menos dos vértices \mathcal{J} -contraíbles.

Evidentemente el Teorema 3.5 de [16] es falso. Para mostrar que el axioma es falso basta tomar a $v = v_1$ y por simetría es suficiente mostrar que las vecindades abiertas $N_G(1, 8)$ y $N_G(1, 9)$ no son \mathcal{J} -contraíbles, como se aprecia en la siguiente figura:



donde ambas vecindades $N_G(1,9)$ y $N_G(1,8)$ poseen un punto aislado, a saber v_{13} y v_{10} respectivamente.

2.3. Gráficas colapsables

La propiedad de una gráfica de ser colapsable está establecida en la categoría de complejos simpliciales.

Un colapso simplicial de Δ en $\tilde{\Delta}$ se obtiene por la remoción de todos los simplejos $\gamma \in \Delta$ tales que $\sigma \subseteq \gamma \subseteq \tau$, siempre que (σ, τ) es una pareja libre, escribiremos $\Delta \searrow \tilde{\Delta}$. Adicionalmente, si $\dim(\tau) = \dim(\sigma) + 1$, entonces $\Delta \searrow \tilde{\Delta}$ es llamado un *colapso simplicial elemental*. No es difícil ver que cualquier colapso simplicial puede ser realizado por colapsos elementales (cf. [11, Sec. III.4])

Si existen $\Delta_1, \Delta_2, \dots, \Delta_n$ complejos simpliciales tales que $\Delta_1 \searrow \Delta_2 \searrow \dots \searrow \Delta_n$, decimos que Δ_1 es colapsable a Δ_n , con $n = 1$ inclusive. Denotamos esto por $\Delta_1 \searrow \Delta_n$. En particular, si Δ es colapsable a Δ^0 decimos que Δ es colapsable. Escribimos $C(\Delta; (\sigma, \tau)) = \Delta - (\sigma, \tau)$ donde (σ, τ) es una pareja libre del complejo simplicial Δ y definimos

$$C(\Delta; ((\sigma_1, \tau_1), (\sigma_2, \tau_2), \dots, (\sigma_k, \tau_k))) = C(C(\Delta; (\sigma_1, \tau_1)); ((\sigma_2, \tau_2), \dots, (\sigma_k, \tau_k)))$$

para una sucesión de colapsos de parejas libres.

Sea $G = (V, E)$ una gráfica simple finita. El complejo simplicial de subgráficas completas $\Delta(G)$ de una gráfica G , es el complejo simplicial (finito) con todas las subgráficas completas de G como simplejos. El 1-esqueleto de $\Delta(G)$ puede ser identificado con G .

Un complejo simplicial Δ es llamado un complejo de *Whitney* si existe una gráfica G tal que $\Delta = \Delta(G)$.

Adicionalmente, requerimos establecer que al complejo de Whitney de una vecindad abierta se le llama *link* de v en $\Delta(G)$ para diferenciar vecindades de v en G de las vecindades de v en $\Delta(G)$,

$$\Delta(N_{G(v)}) = \text{Lk}(v; \Delta(G)).$$

Definición 2.3.1. Denotemos como \mathfrak{C} a la familia de gráficas G tales que $\Delta(G)$ es colapsable. Una gráfica $G \in \mathfrak{C}$ es llamada gráfica colapsable.

Por ejemplo, cualquier gráfica $K(n)$ es colapsable: $\Delta(K(n)) \simeq \Delta^0$.

En [24] se probó que cualquier gráfica desmantelable es contraíble. En [5] esta familia fue extendida a gráficas s -desmantelable. Más aún, fueron definidos los s -movimientos, y se probó que los s -movimientos no cambian la homología de la gráfica. En particular se probó que las gráficas s -desmatelables tienen homología trivial.

En [18] existe un estudio a cerca de la relación del k -**comportamiento** respecto a la homología. En [17] se llevó a cabo un estudio de la trivialidad de los grupos de la homología de complejos de completas aleatorios.

2.4. Comparación de familias fuertemente \mathfrak{J} -contraíbles y colapsables

Obtener ejemplos de gráficas pequeñas de la familia \mathfrak{J}_f no es una tarea difícil, llama la atención que se pueda verificar que las gráficas obtenidas también son colapsables. Es precisamente lo que afirma el siguiente teorema.

Teorema 2.4.1. Si G es una gráfica fuertemente \mathfrak{J} -contraíble, entonces $\Delta(G)$ es un complejo simplicial colapsable.

Demostración. Para demostrar este resultado debemos considerar una gráfica G perteneciente a la familia \mathfrak{J}_f , es decir, que sea contraíble mediante las transformaciones de Ivashchenko (I1) y (I3), y demostrar que el complejo simplicial asociado $\Delta(G)$ es colapsable ($G \in \mathfrak{C}$), esto es, que existe una sucesión de colapsos (podemos considerarlos elementales) que reduce $\Delta(G)$ al complejo simplicial trivial Δ^0 .

La demostración que a continuación se presenta es dada por inducción sobre la dimensión n del complejo simplicial $\Delta(G)$; equivalentemente, sobre la existencia de una subgráfica maximal $K(n+1)$ de G .

Recordemos que una gráfica es fuertemente \mathfrak{J} -contraíble si y sólo si, es posible \mathfrak{J} -contraer la gráfica usando exclusivamente borrados de vértices o aristas con vecindades en la familia \mathfrak{J}_f .

Establecemos para cada $n \geq 0$, la siguiente proposición:

$$\boxed{P(n) : \text{Si } \dim(\Delta(G)) = n, \text{ entonces: } G \in \mathfrak{J}_f \Rightarrow G \in \mathfrak{C}.}$$

De modo que el teorema que deseamos probar afirma la validez de $P(n)$ para todo $n \geq 0$.

I) $P(0)$: Este caso es trivial.

II) Hipótesis de inducción: Sea $n \geq 1$ y supongamos que $P(m)$ es verdadero para todo $m \leq n - 1$.

III) Procedamos ahora a demostrar, bajo la hipótesis anterior, que $P(n)$ es válido. En otras palabras, debemos demostrar que para toda gráfica $G \in \mathfrak{J}_f$ de dimensión n , su complejo simplicial asociado $\Delta(G)$ es colapsable.

Es posible que este punto (III) pueda ser demostrado en diversas maneras. La técnica que usaremos para probar (III) es por inducción sobre el número de vértices y aristas de G .

Sea $k = |V(G)| + |E(G)|$ el número total de vértices y aristas de la gráfica G , cuyo complejo simplicial asociado $\Delta(G)$ tiene dimensión n .

Establecemos para cada $k \geq 1$, la siguiente proposición:

$Q(k)$: Sea G una gráfica tal que $\dim(\Delta(G)) = n$. Si $|V(G)| + |E(G)| = k$, entonces $\Delta(G)$ es colapsable.

Es claro que esta proposición tiene sentido sólo para $k \geq n + 1 + \binom{n+1}{2}$, pues $K(n+1)$ es la gráfica más pequeña de dimensión n y además: $|V(K(n+1))| = n+1$, $|E(K(n+1))| = \binom{n+1}{2}$.

Nuestro objetivo ahora es demostrar que $Q(k)$ es válido para todo $k \geq n + 1 + \binom{n+1}{2}$. Pues al demostrar esto, tendremos en particular que $P(n)$ es cierto.

Para demostrar la validez de $Q(k)$ para todo $k \geq n + 1 + \binom{n+1}{2}$, procedemos por inducción.

1) $Q(n + 1 + \binom{n+1}{2})$: Este caso es trivial. La única gráfica G de dimensión $\dim(\Delta(G)) = n$ y con $n + 1 + \binom{n+1}{2}$ vértices y aristas es $G = K(n + 1)$. Y es claro que $K(n + 1)$ es colapsable.

2) Hipótesis de inducción: Supongamos que $Q(j)$ es verdadero para todo $n + 1 + \binom{n+1}{2} \leq j \leq k - 1$.

3) Procedamos ahora a demostrar, bajo la hipótesis anterior, que $Q(k)$ es válido. Así, sea $G \in \mathfrak{J}_f$ una gráfica tal que $\dim(\Delta(G)) = n$ y que tiene k vértices y aristas. Demostremos que $\Delta(G)$ es colapsable.

Sea α vértice o arista de G , como $G \in \mathfrak{J}_f$, se sigue que existe $\alpha \in G$ tal que $N_G(\alpha) \in \mathfrak{J}_f$, es decir, que su vecindad pertenece a la familia de graficas fuertemente \mathfrak{J} -contraíbles.

Por otro lado, como el complejo asociado a esta vecindad satisface que $\Delta(N_G(\alpha)) = \text{Lk}(\alpha; \Delta(G))$, se sigue que su dimensión cumple

$$\dim(\Delta(N_G(\alpha))) = \dim(\text{Lk}(\alpha; \Delta(G))) \leq n - 1.$$

Luego, como toda gráfica $H \in \mathfrak{J}_f$ cuyo complejo simplicial $\Delta(H)$ es de dimensión menor o igual que $n - 1$ es colapsable (hipótesis global: $P(m)$, $\forall m \leq n - 1$),

entonces se sigue que la gráfica \mathfrak{J}_f -contraíble $H = N_G(\alpha)$ es colapsable a un vértice $w \in \Delta(N_G(\alpha))$.

Luego, como el complejo simplicial $\Delta(N_G(\alpha))$ es colapsable, existe una sucesión de colapsos elementales que lo reducen simplicialmente al complejo trivial,

$$\Delta(N_G(\alpha)) \searrow \cdots \searrow \Delta^0.$$

Cada uno de dichos colapsos elementales es dado por el colapso elemental de una pareja libre:

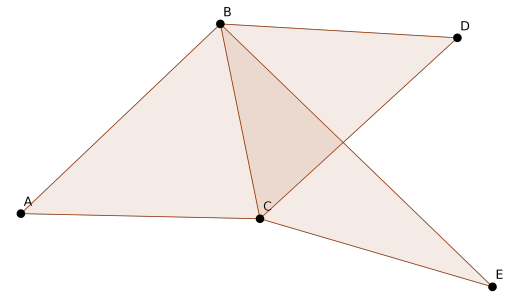
$$\Delta^0 = (\cdots((\Delta(N_G(\alpha)) - (\sigma_1, \tau_1)) - (\sigma_2, \tau_2))\cdots),$$

esto es, $\dim(\tau_i) = \dim(\sigma_i) + 1$, para cada $i \geq 1$.

Por otro lado, como $\alpha \notin \Delta(N_G(\alpha))$ se sigue que si (σ_i, τ_i) es una pareja libre en $\Delta(N_G(\alpha))$, entonces $(\alpha * \sigma_i, \alpha * \tau_i)$ también es pareja libre en $\alpha * \text{Lk}(\alpha; \Delta(N_G(\alpha)))$. De hecho, $(\alpha * \sigma_i, \alpha * \tau_i)$ es pareja libre en $\Delta(G)$.

Esto último que se comenta, no sucede en general. Por ejemplo, el vértice C es una cara libre del complejo simplicial dado por la arista $\{A, C\}$; además, el vértice B es un vértice ajeno a $\{A, C\}$ y entonces $(B * C, B * \{A, C\})$ es una pareja libre en la estructura simplicial $B * \{A, C\} = \{A, B, C\}$. Sin embargo, es evidente que $(B * C, B * \{A, C\})$ no es una pareja libre en la estructura sim-

plicial completa de la imagen de la derecha.

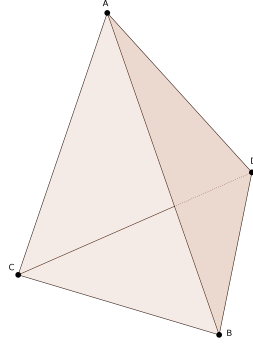


Por otro lado, y en nuestro caso, si (σ_i, τ_i) es una pareja libre en $\Delta(N_G(\alpha))$ y $(\alpha * \sigma_i, \alpha * \tau_i)$ no es pareja libre en $\Delta(G)$ (para algún $i \geq 1$), entonces existe un simplejo $\gamma \in \Delta(G)$, $\gamma \neq \alpha * \tau_i$, con $\dim(\gamma) \geq \dim(\alpha * \sigma_i) + 1$ y $\alpha * \sigma_i \leq \gamma$. Esto último implica que $V(\gamma) \setminus V(\alpha) \subset V(N_G(\alpha))$. Sea $\tilde{\alpha} \subset V(\gamma) \setminus V(\alpha * \tau_i)$ no vacío, se sigue que $\tilde{\alpha} * \sigma_i$ es un simplejo de dimensión $\dim(\tilde{\alpha} * \sigma_i) \geq 1 + \dim(\sigma_i)$ y que contiene a σ_i ; esto contradice el hecho de que (σ_i, τ_i) es pareja libre en $\Delta(N_G(\alpha))$.

Aplicando cada uno de los colapsos correspondientes a las parejas libres $(\alpha * \sigma_i, \alpha * \tau_i)$ en $\Delta(G)$, y posteriormente el colapso asociado a la pareja libre $(\alpha, \alpha * w)$, obtenemos un colapso de $\Delta(G)$ en $\Delta(G - \alpha)$. Finalmente, como $\Delta(G - \alpha)$ tiene a lo más $k-1$ vértices y aristas, por hipótesis de inducción sobre k (es válido $Q(j), \forall j \leq k-1$), tenemos que el complejo simplicial $\Delta(G - \alpha)$ es colapsable cuando $G - \alpha \in \mathfrak{J}_f$.

□

Observación. 2.4.2. En general, si Δ es un complejo simplicial y $G_\Delta = \Delta^{(1)}$ su gráfica subyacente (1-esqueleto), entonces no necesariamente se tiene que $\Delta(N_{G_\Delta}(\alpha)) = \text{Lk}(\alpha; \Delta)$. Por ejemplo, si Δ consiste del complejo simplicial obtenido al remover una cara y el interior del tetraedro,



entonces su gráfica subyacente es $G_\Delta = K(4)$ y,

$$\Delta(N_{G_\Delta}(A)) = [B, C, D] \neq \{[C, D], [B, D]\} = \text{Lk}(A; \Delta)$$

En general, dado un vértice $v \in \Delta$ en el complejo simplicial Δ , se tiene la contención:

$$\begin{aligned} \text{Lk}(v; \Delta) &:= \{\sigma \in \Delta \mid \{v\} \cap \sigma = \emptyset, \{v\} \cup \sigma \in \Delta\} \\ &\subseteq \{\sigma \subset N_{G_\Delta}(v) \mid \{v\} \cap \sigma = \emptyset\} \end{aligned}$$

Corolario 2.4.3 ([15, Teo. 4.9]). El pegado o borrado de vértices fuertemente \mathfrak{J} -contraíbles no cambia los grupos de homología de una gráfica.

Demostración. Sea G una gráfica finita simple y $G_0 \subset G$ una subgráfica fuertemente \mathfrak{J} -contraíble. Del Teorema 2.4.1 se sigue que G_0 es una gráfica colapsable y por lo tanto, $v * G_0$ también es colapsable para cualquier vértice $v \notin G$. En consecuencia, agregar o remover vértices fuertemente \mathfrak{J} -contraíbles corresponden a colapsos simpliciales mediante parejas libres en el complejo de Whitney $\Delta(G)$, los cuales preservan los grupos de homología simplicial (ver [14, Cap. 2] y Sección 3.2). \square

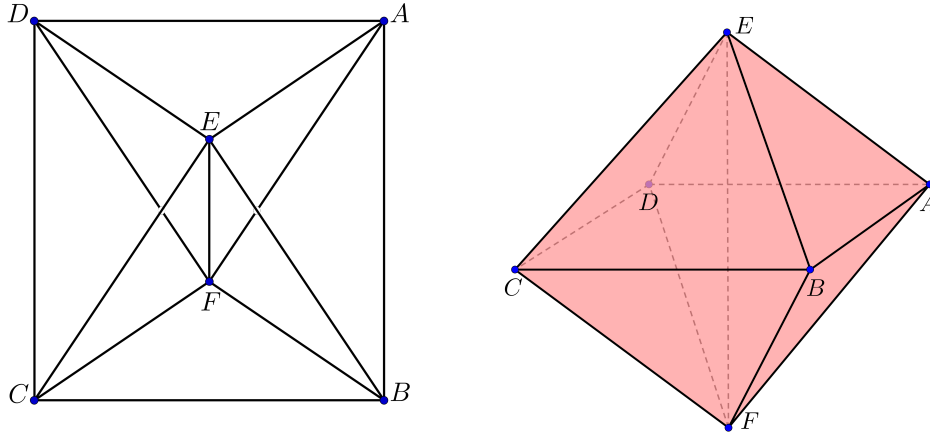
Hemos probado que $\mathfrak{J}_f \subseteq \mathfrak{C}$ a través de verificar que las operaciones \mathfrak{J} -contraíbles I2 e I4 en G , inducen una sucesión de colapsos elementales de $\Delta(G)$ en Δ^0 . Adicionalmente establecemos la siguiente afirmación:

Conjetura: La familia de gráficas fuertemente \mathfrak{J} -contraíbles es igual a la familia de gráficas colapsables.

Sin embargo, para verificar que $\mathfrak{C} \subseteq \mathfrak{J}_f$, no siempre es posible utilizar como guía la sucesión de colapsos de parejas libres, para realizar las \mathfrak{J}_f -contracciones.

Ejemplo 2.4.4. Sea G la gráfica de la izquierda en la Figura 2.1. El complejo de completas $\Delta(G)$ está a la derecha de la imagen.

Claramente $G \in \mathfrak{J}_f$, y $\Delta(G)$ es colapsable, i.e. $G \in \mathfrak{C}$. En sentido inverso los argumentos en el Teorema 2.4.1, cualquier secuencia de eliminación de parejas libres debe inducir una secuencia de transformaciones contraíbles. Los colapsos, sin embargo, no pueden inducir una transformación contraíble:


 Figura 2.1: La gráfica G y su complejo de Whitney $\Delta(G)$.

Removemos el tetraedo $ABEF$ y su cara libre ABE , después removemos el relleno de los triángulos internos del octaedro: AEF, DEF, CEF y por último la arista $\{E, F\}$, en símbolos:

$$\begin{aligned} \Delta_0 &:= \Delta(G) \searrow \Delta_1 := C(\Delta_0; (\{A, B, E\}, \{A, B, E, F\})) \\ &\searrow \Delta_2 := C(\Delta_1; (\{A, E, F\}, \{A, D, E, F\})) \\ &\searrow \Delta_3 := C(\Delta_2; (\{D, E, F\}, \{C, D, E, F\})) \\ &\searrow \Delta_4 := C(\Delta_3; (\{C, E, F\}, \{B, C, E, F\})) \\ &\searrow \Delta_5 := C(\Delta_4; (\{E, F\}, \{B, E, F\})). \end{aligned}$$

De hecho, el 1-esqueleto de Δ_5 no es una gráfica contraíble, y $N_G(\{E, F\}) \notin \mathcal{J}_f$.

Observación. 2.4.5. En [5], los autores presentan el concepto de ws-desmantelable; ellos afirman que un vértice v en una gráfica G es ws-desmantelable si $N(v)$ es desmantelable, en este caso escriben $ws(G; v) = G - v$. Una arista $\{u, v\}$ en la gráfica G es llamada ws-desmantelable si $N_G(u) \cap N_G(v)$ es desmantelable, en este caso escribimos $ws(G; \{u, v\}) = G - \{u, v\}$. Obviamente $\{E, F\}$ no es ws-desmantelable en Δ_4 .

Así, un colapso de una pareja libre no es, en general, una transformación contraíble o ws-desmantelable. Compárese esta observación con la prueba del Lema 4.4 de [5].

Apesar del Ejemplo 2.4.4, hemos conjeturado que cualquier gráfica colapsable es también una gráfica fuertemente \mathcal{J} -contraíble. En esta sección mostramos algoritmos que hemos utilizado para verificar la inclusión $\mathcal{C} \subset \mathcal{J}_f$ para un conjunto finito de gráficas. Estos algoritmos fueron escritos en C/C++ están disponibles en el repositorio [3].

Para apoyar la conjetura ($\mathcal{C} \subset \mathcal{J}_f$), se ha calculado un subconjunto de estas familias de gráficas, restringiendo el número de vértices a $n \leq 9$.

Dado que la conectividad es una característica común a estas familias, la primera estrategia es obtener una colección de gráficas conexas, entonces para cada gráfica se aplicarán

dos algoritmos para determinar si $G \in \mathcal{J}_f$ y si $G \in \mathcal{C}$. En caso de obtener una gráfica $G \in \mathcal{C}$ y que en un primer intento aparentemente $G \notin \mathcal{J}_f$, se requeriría encontrar una secuencia ordenada de \mathcal{J} -contracciones que transformen a G en $K(1)$. Los programas y sus resultados se encuentran en el repositorio antes mencionado.

2.5. Construcción iterada de gráficas fuertemente \mathcal{J} -contraíbles

En seguida se muestra un algoritmo recursivo para determinar si una gráfica conexa puede \mathcal{J}_f -contraerse hasta la gráfica $K(1)$, mediante sólo la operación de borrado de vértices \mathcal{J}_f -contraíbles.

Como se mostró anteriormente, existen gráficas que no son \mathcal{J}_f -contraíbles a $K(1)$ sólo mediante el borrado de vértices \mathcal{J} -contraíbles. Sin embargo, la evidencia computacional muestra que para gráficas de un número pequeño de vértices, estas dos familias son de tamaño similar y además, la operación de \mathcal{J} -contraer vértices presenta propiedades importantes en el cálculo de homología persistente de ciertas estructuras simpliciales (complejo de Vietoris-Rips), como veremos en la siguiente sección.

El algoritmo verifica que una gráfica puede ser reducida a un punto sólo utilizando la operación de borrado de vértices. La única gráfica que se da por conocida de la familia \mathcal{J}_f es $K(1)$.

El algoritmo propuesto da por hecho que se tienen funciones o procedimientos para calcular la vecindad abierta de un vértice y también para remover un vértice de la gráfica.

Algorithm 3: `vertex.contractible.graph`

Input : Una grafica G y la cardinalidad n del conjunto de vértices.

Output: El valor lógico TRUE si $G \in \mathcal{J}_f$, o FALSE en otro caso.

```

1 if  $n = 0$  then
2   | return FALSE;
3 else
4   | if  $n = 1$  then
5     | return TRUE;
6   | else
7     | for  $i \leftarrow 1$  to  $n$  do
8       | if vertex.contractible.graph( $N_G(v_i), k$ ) = TRUE then
9         | | return vertex.contractible.graph( $G - v_i, n - 1$ )
10        | end
11       | end
12     | return FALSE;
13   | end
14 end

```

Observación. 2.5.1. El algoritmo anterior nos dice si es posible contraer una gráfica sin ser exhaustivo en su búsqueda, es decir, no intenta todos los órdenes posibles para eliminar vértices \mathcal{J} -contraíbles. Por este motivo no garantiza obtener la familia completa de gráficas fuertemente contraíbles por eliminación sólo de vértices.

Para decidir qué vértice es el siguiente, se utiliza un esquema de prioridades, esto es, siempre se le da prioridad a los primeros vértices de ser eliminados. Cada vez que un vértice no puede ser eliminado, se intenta con el siguiente vértice. En caso de poder eliminar un vértice, se vuelve a intentar con los primeros vértices.

Aun con el esquema anterior de prioridades, no podemos garantizar obtener la familia completa. Sin embargo, hasta donde se ha realizado la experimentación, no ha sido necesaria una búsqueda exhaustiva a fin de determinar si es posible contraer una gráfica eliminando sólo vértices.

2.5.1. Algoritmo de reducción por \mathcal{I} -contracción de vértices

Muchas gráficas conexas no pertenecen a la familia \mathcal{I}_f ; sin embargo, es deseable determinar en qué medida es posible eliminar vértices de la gráfica G .

Algorithm 4: `vertex.contractible.reduction`

Input : Una gráfica finita G .
Output: Una gráfica reducida $I(G; S)$ y un conjunto ordenado de vértices S .

```

1 reduced ← FALSE;
2 S ← ∅;
3 while reduced = FALSE do
4   reduced ← TRUE ;
5   for v ∈ V(G) do
6     if vertex.contractible.graph(N_G(v), k) = TRUE then
7       reduced ← FALSE;
8       actualizar gráfica G ← G - v;
9       S ← S ∪ {v};
10      break for
11    end
12  end
13 end
14 return (G, S)

```

El algoritmo `vertex.contractible.reduction` devuelve una gráfica maximalmente reducida respecto al borrado de vértices. Sin embargo, es posible reducir la gráfica $I(G; S)$ utilizando borrado de aristas \mathcal{I}_f -contraíbles como se muestra en la Figura 2.2, por lo pronto aplicamos sólo el borrado de vértices, toda vez que es una operación que realiza modificaciones más relevantes en la gráfica, cada vez que se elimina un vértice, también se eliminan todas las aristas que en él inciden.

Para la gráfica en la Figura 2.2, tenemos $I(G; S) = G$, i.e. $S = \emptyset$. La arista $\{v_1, v_2\}$, sin embargo, puede ser borrada por la transformación \mathcal{I} -contraíble ($\mathcal{I}\mathcal{B}$), incluso cuando no es posible eliminar ningún vértice de G a través de una transformación \mathcal{I} -contraíble.

Observación. 2.5.2. Las familias \mathcal{I}_f y \mathcal{C} son infinitas; sin embargo, se pueden calcular subfamilias para hacer la comparación. En el repositorio [3] están los códigos en C/C++ para calcular subfamilias limitadas a nueve vértices, así como sus matrices de adyacencia y algunas representaciones geométricas. Para calcular estas subfamilias, necesitamos el algoritmo para obtener la subfamilia \mathcal{C} , el cual se realiza por simple inspección; sin embargo, también proporcionamos el código. A continuación describimos brevemente el algoritmo, que dado su baja eficacia, sólo damos generalidades.

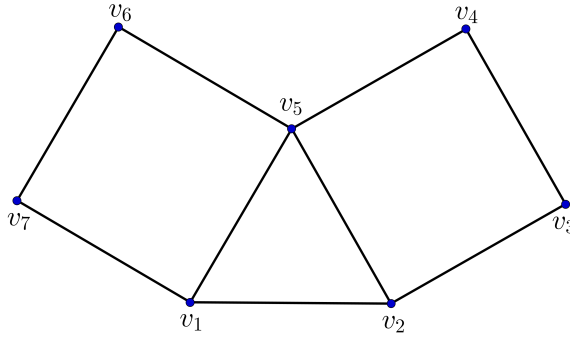


Figura 2.2: Gráfica \mathfrak{J} -contraíble-reducida con una arista \mathfrak{J} -contraíble.

Algoritmo para determinar si una gráfica es colapsable.

La idea general del algoritmo es determinar si $\Delta(G)$ se puede colapsar a $K(1)$. El algoritmo se basa en los siguientes procedimientos:

- Cálculo de subgráficas completas.
- Cálculo de matriz de contención.
- Detección de parejas libres.
- Eliminación de parejas libres.

Obtener de manera eficiente las subgráficas completas de una gráfica G , no es una tarea simple. El algoritmo implementado en el repositorio examina la matriz de adyacencia M para determinar que elementos de 2^V también son elementos de G .

La matriz de contención almacena información binaria que indica si la subgráfica completa i esta contenida en la subgráfica completa j . En base a esta información se realiza la detección de parejas libres. La búsqueda de parejas libres le da prioridad a parejas libres de mayor dimensión y que contienen mayores índices.

Cada vez que se elimina una pareja libre, se actualiza la lista de subgráficas completas, así como la matriz de contención. En dado caso que no se detecten parejas libres y que aun no se haya colapsado $\Delta(G)$ a $K(1)$, se decide informar que la gráfica no es colapsable.

Observación. 2.5.3. Esta versión del algoritmo no garantiza obtener un subconjunto completo de gráficas colapsables, pues no examina todas las posibles maneras de realizar los colapsos, en dado caso que no pueda colapsar en su primer intento. De detectarse una gráfica G que difiere respecto a la pertenencia a la familia \mathfrak{J}_f , se debe realizar la revisión exhaustiva, hasta el momento y para gráficas pequeñas no ha sido necesario intentar todos los posibles órdenes para colapsar.

2.6. Gráficas homotópicamente colapsables

Por lo menos existen dos gráficas que no son colapsables pero que si son \mathfrak{J} -contraíbles. A saber la *casa de Bing* y *dunce hat*. En ambos casos las gráficas pueden ser transformadas

a una gráfica completa con la operación [I4] (pegado de aristas \mathfrak{J} -contraíbles) y de ahí pueden ser transformadas a $K(1)$ mediante \mathfrak{J} -contracciones.

Ver [8] para la afirmación anterior sobre la casa de Bing. Para verificar la secuencia para transformar a la gráfica dunce hat (doble división baricéntrica) en la gráfica $K(17)$ remitimos al Apéndice B.

El hecho de que en ambas gráficas se requiera agregar aristas, para poder luego contraer a $K(1)$, nos lleva a considerar la inclusión de la operación inversa a los colapsos.

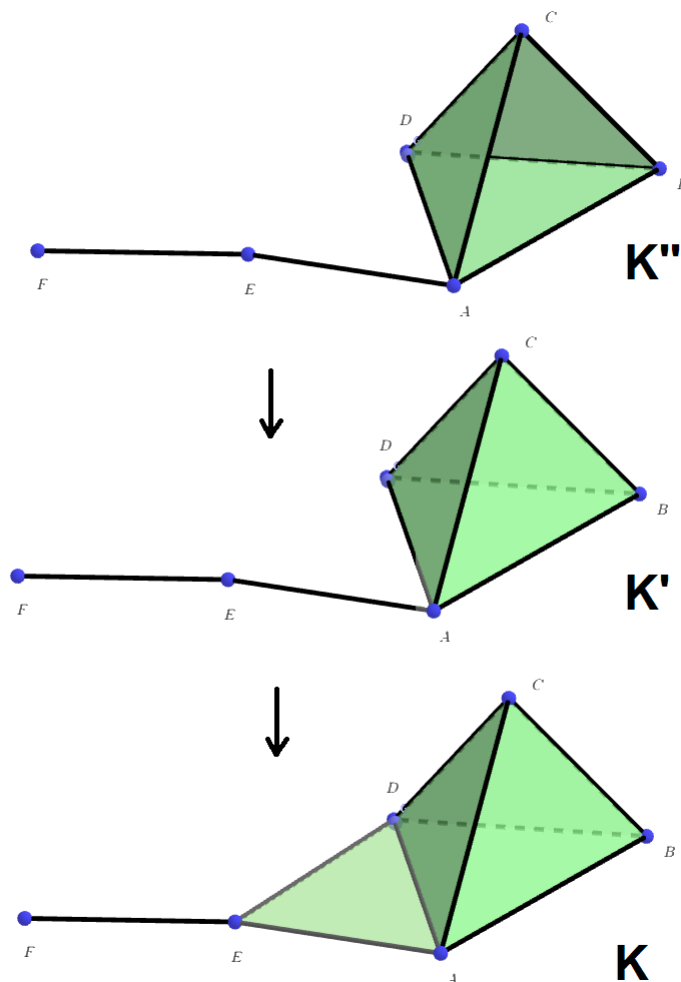
Definición 2.6.1. Sea (σ, τ) una pareja libre en el complejo simplicial K y $K' = K \setminus (\sigma, \tau)$ con σ de dimensión n y τ de dimensión $n - 1$. A la operación se le llama colapso elemental de K a K' o **expansión elemental** de K' a K .

Ejemplo 2.6.2. Al los complejos simpliciales representados en la siguiente figura aplicamos colapsos elementales y expansiones elementales.

$$K' = K \setminus (\{A, D, E\}, \{D, E\}), K \searrow K',$$

$$K'' = K' \setminus (\{A, B, C, D\}, \{A, B, C\}), K'' \searrow K',$$

Así $K'' \nearrow K'$ y $K' \nearrow K$ mediante expansiones elementales



Definición 2.6.3. Si el complejo simplicial K se puede obtener a partir de $K(1)$ mediante colapsos elementales y/o expansiones elementales le llamaremos complejo **homotópicamente colapsable**. Si además $K = \Delta(G)$, a G le llamaremos gráfica homotópicamente colapsable ($\mathfrak{H}\mathfrak{C}$).

Establecemos la siguiente conjetura: *La familia \mathfrak{I} -contraíble coincide con la familia de gráficas homotópicamente colapsables.*

Capítulo 3

Propiedades homológicas de gráficas fuertemente \mathfrak{J} -contraíbles

3.1. Homología de gráficas

Sea \mathfrak{G} el conjunto de todas la gráficas simples y sea $G = (V, E) \in \mathfrak{G}$. Para definir la homología de una gráfica utilizamos la notación de [15, Sec. 3]; lo cual de hecho, es equivalente a definir la homología de la gráfica G como la homología simplicial del complejo de Whitney $\Delta(G)$.

Sea σ_n una gráfica completa (isomorfa a $K(n+1)$), cuyo conjunto de vértices es $\{v_0, \dots, v_n\}$. Una orientación para la gráfica σ_n consiste en fijar un orden arbitrario para sus vértices, salvo permutaciones pares; esto es, corresponde a una clase de equivalencia del conjunto cociente $\{v_0, \dots, v_n\}/\mathcal{S}_n$, inducida por la acción por permutaciones del grupo simétrico \mathcal{S}_n sobre el conjunto de vértices.

La frontera orientada $\partial\sigma^n$ de la gráfica completa σ^n está definida como una combinación lineal (formal) de sus subgráficas completas de n vértices:

$$\partial\sigma^n = \partial[v_0v_1\cdots v_n] = \sum_{k=0}^n (-1)^k [v_0v_1\cdots\hat{v}_k\cdots v_n],$$

donde la notación \hat{v}_k indica que el vértice v_k debe ser omitido.

Sea A un grupo abeliano, el cual usualmente se toma como el grupo de los números enteros o algún campo finito. Una n -cadena de la gráfica G , con coeficientes en A , está definida formalmente como la combinación lineal (formal) de subgráficas completas σ^n de la gráfica G :

$$c_n = \sum_k \alpha_k \sigma_k^n,$$

para $\alpha_k \in A$. La suma de n -cadenas está definida de manera obvia. Tenemos entonces el grupo de cadenas $C_n(G)$ de todas las n -cadenas en G .

Se define el operador frontera $\partial_n : C_n(G) \rightarrow C_{n-1}(G)$ por extensión lineal. La frontera de una 0-cadena está definida como cero. Se puede probar directamente que $\partial_n \circ \partial_{n+1} = 0$, entonces tenemos un complejo de cadenas:

$$\cdots \xrightarrow{\partial_{n+1}} C_n(G) \xrightarrow{\partial_n} C_{n-1}(G) \xrightarrow{\partial_{n-1}} \cdots \xrightarrow{\partial_2} C_1(G) \xrightarrow{\partial_1} C_0(G) \longrightarrow 0.$$

El grupo de homología n -dimensional de la gráfica G , con coeficientes en A , está definido por

$$H_n(G; A) = \frac{\ker (\partial_n : C_n(G) \rightarrow C_{n-1}(G))}{\text{Im} (\partial_{n+1} : C_{n+1}(G) \rightarrow C_n(G))}.$$

Las n -cadenas en el subgrupo $Z_n(G) := \ker (\partial_n : C_n(G) \rightarrow C_{n-1}(G))$ se llaman n -ciclos, y las n -cadenas en el subgrupo $B_n(G) := \text{Im} (\partial_{n+1} : C_{n+1}(G) \rightarrow C_n(G))$ se llaman n -fronteras.

Identificando cada subgráfica completa $\sigma_n \subset G$ de la gráfica G , con el correspondiente simplejo del complejo de Whitney $\Delta(G)$, obtenemos un isomorfismo canónico entre los grupos de homología de la gráfica G y los grupos de homología del complejo simplicial $\Delta(G)$, i.e.

$$H_*(G; A) \cong H_*^\Delta(\Delta(G); A).$$

Escribimos $H_*^\Delta(-)$ para denotar el funtor de homología simplicial, con el fin de diferenciarlo de la homología de la gráfica. Omitiremos en el resto del trabajo, el grupo de coeficientes en la notación del grupo de homología.

3.2. Invariancia homológica del borrado de vértices fuertemente \mathfrak{J} -contraíbles

Notación. 3.2.1. Denotamos por $I(G; v) = G - v$ a la gráfica obtenida a partir de G , al borrar un vértice \mathfrak{J}_f -contraíble, esto es, $N_G(v)$ pertenece a la familia \mathfrak{J}_f .

Sea $S = (v_1, \dots, v_k) \subset V$ un conjunto ordenado no vacío de vértices de la gráfica $G = (V, E)$. Decimos que S es una *sucesión de vértices \mathfrak{J}_f -contraíbles* de G si v_1 es un vértice \mathfrak{J}_f -contraíble de $G_1 := G$ y, si $k \geq 2$, entonces v_i es \mathfrak{J}_f -contraíble en $G_i := I(G_{i-1}; v_{i-1})$ para todo $2 \leq i \leq k$.

Si $S = (v_1, \dots, v_k)$ es una sucesión de vértices \mathfrak{J}_f -contraíbles de la gráfica G , definimos $I(G; S) := I(I(G; v_1); (v_2, \dots, v_k))$.

En [15, Th. 4.9], se probó que la transformación \mathfrak{J}_f -contraíble $G \mapsto I(G; v)$ no cambia los grupos de homología de la gráfica G . En consecuencia, se tiene un isomorfismo $H_*(G) \cong H_*(I(G; S))$ para toda sucesión de vértices \mathfrak{J}_f -contraíbles de G .

El isomorfismo inducido $I_*(v) : H_n(G) \rightarrow H_n(I(G; v))$ se puede dar explícitamente como sigue. Sea $c_n \in C_n(G)$ una n -cadena de la gráfica G ; entonces tenemos que $c_n = a_n + v * b_{n-1}$ para $a_n \in C_n(I(G; v))$ y $b_{n-1} \in C_{n-1}(N_G(v))$. Si $c_n \in Z_n(G)$, entonces $\partial c_n = \partial a_n + b_{n-1} - v * \partial b_{n-1} = 0$ implica $\partial a_n + b_{n-1} = 0$ y $\partial b_{n-1} = 0$.

De la contractibilidad de $N_G(v)$, existe una n -cadena $b_n \in C_n(N_G(v))$ tal que $\partial b_n = b_{n-1}$. Tenemos que

$$c_n = a_n + v * \partial b_n \quad \text{y} \quad \partial(a_n + b_n) = 0.$$

Dado que $\partial(v * b_n) = b_n - v * \partial b_n$, la ecuación de arriba puede escribirse como

$$c_n = (a_n + b_n) - \partial(v * b_n).$$

El isomorfismo $I_*(v) : H_n(G) \rightarrow H_n(I(G; v))$ es inducido por el homomorfismo

$$I_{\#}(v) : Z_n(G) \rightarrow Z_n(I(G; v)), (a_n + b_n) - \partial(v * b_n) \mapsto a_n + b_n.$$

Para una sucesión de vértices \mathfrak{J}_f -contraíbles $S = (v_1, v_2, \dots, v_k)$, tenemos una colección de homomorfismos inducidos en la colección de ciclos:

$$Z_*(G) \xrightarrow{I_\#(v_1)} Z_*(G - \{v_1\}) \xrightarrow{I_\#(v_2)} \dots \xrightarrow{I_\#(v_k)} Z_*(G - \{v_1, v_2, \dots, v_k\}).$$

Denotamos la composición de tales homomorfismos por $I_\#(S)$, i.e.

$$I_\#(S) = I_\#(v_1, v_2, \dots, v_k) := I_\#(v_k) \circ \dots \circ I_\#(v_2) \circ I_\#(v_1),$$

Asimismo, denotaremos por $I_*(S) : H_*(G) \rightarrow H_*(I(G; S))$ el homomorfismo que induce en homología el morfismo $I_\#(S) : Z_*(G) \rightarrow Z_*(I(G; S))$.

Teorema 3.2.2. Sea G_1 una gráfica finita y sea $G_0 \subset G_1$ una subgráfica. Si S_0 y S_1 son sucesiones de vértices \mathfrak{J}_f -contraíbles de G_0 y G_1 , respectivamente, entonces existe un homomorfismo $\iota_* : H_*(I(G_0; S_0)) \rightarrow H_*(I(G_1; S_1))$ tal que el siguiente diagrama conmuta:

$$\begin{array}{ccc} H_*(G_0) & \xrightarrow{i_*} & H_*(G_1) \\ I_*(S_0) \downarrow \cong & & I_*(S_1) \downarrow \cong \\ H_*(I(G_0; S_0)) & \xrightarrow{\iota_*} & H_*(I(G_1; S_1)). \end{array} \quad (3.1)$$

Demostración. Sean $S_0 = (v_1, \dots, v_m)$ y $S_1 = (w_1, \dots, w_n)$ sucesiones de vértices \mathfrak{J}_f -contraíbles de las gráficas G_0 y G_1 , respectivamente. Cada una de estas sucesiones induce una sucesión de homomorfismos en los grupos de ciclos de las gráficas G_0 y G_1 , como se ilustra en el siguiente diagrama:

$$\begin{array}{ccc} Z_*(G_0) \xrightarrow{I_\#(v_1)} Z_*(I(G_0; v_1)) \xrightarrow{I_\#(v_2)} \dots \xrightarrow{I_\#(v_m)} Z_*(I(G_0; S_0)) \\ i_\# \downarrow \\ Z_*(G_1) \xrightarrow{I_\#(w_1)} Z_*(I(G_1; w_1)) \xrightarrow{I_\#(w_2)} \dots \xrightarrow{I_\#(w_n)} Z_*(I(G_1; S_1)). \end{array}$$

La flecha vertical está inducida por la inclusión $i : G_0 \hookrightarrow G_1$. Además, los renglones en el diagrama no necesariamente tienen igual longitud.

Se demuestra en [15] que cada flecha horizontal del diagrama es un isomorfismo, entonces $I_\#(S_0) := I_\#(v_m) \circ \dots \circ I_\#(v_1)$ y $I_\#(S_1) := I_\#(w_n) \circ \dots \circ I_\#(w_1)$ también son isomorfismos.

Sea $\iota_\# : Z_*(I(G_0; S_0)) \rightarrow Z_*(I(G_1; S_1))$ definido por $z \mapsto I_\#(S_1) \circ i_\# \circ I_\#(S_0)^{-1}(z)$. Así, tenemos los dos diagramas conmutativos siguientes:

$$\begin{array}{ccc} Z_*(G_0) \xrightarrow{I_\#(v_1)} Z_*(I(G_0; S_0)) & & B_*(G_0) \xrightarrow{I_\#(v_1)} B_*(I(G_0; S_0)) \\ i_\# \downarrow & \downarrow \iota_\# & i_\# \downarrow \\ Z_*(G_1) \xrightarrow{I_\#(w_1)} Z_*(I(G_1; S_1)) & & B_*(G_1) \xrightarrow{I_\#(w_1)} B_*(I(G_1; S_1)). \end{array}$$

El diagrama de la izquierda es conmutativo por construcción, y el diagrama de la derecha es conmutativo por restricción a los subgrupos de fronteras. Tomando el morfismo inducido en los grupos cociente (grupos de homología), obtenemos el diagrama conmutativo (3.1). ■

3.3. Aplicación: Homología persistente del complejo de Vietoris-Rips

En algunas aplicaciones de topología algebraica, como el análisis topológico de datos (ATD), se estudia la *forma de una nube de puntos* a través de la homología persistente (cf. [7, 23]).

La estructura simplicial apropiada y los parámetros o variables para construir una filtración sobre una nube de puntos son pasos clave en ATD. Podemos encontrar varias estructuras de complejos simpliciales usualmente empleados en ATD. Una estructura simplicial muy frecuente con la cual se define la geometría sobre una nube de puntos es conocida como el complejo simplicial de Vietoris-Rips.

Sea N una nube de puntos finita en algún espacio métrico (\mathcal{M}, d) , y sea ε un número real no negativo. El complejo de Vietoris-Rips $\text{VR}(N; \varepsilon)$ es un complejo simplicial abstracto con conjunto de vértices N , y $\sigma \subset N$ es un simplejo si y sólo si, las distancias entre cualquier par de puntos en σ es menor o igual que ε :

$$\sigma = \{v_0, v_1, \dots, v_k\} \in \text{VR}(N; \varepsilon) \iff d(v_i, v_j) \leq \varepsilon, \forall v_i, v_j \in \sigma.$$

Claramente, $\text{VR}(N; \varepsilon) \subset \text{VR}(N; \varepsilon')$ para todo $\varepsilon \leq \varepsilon'$. Tomando el conjunto de valores $d(u, v)$ para todo $u, v \in N$ y ordenándolos en forma creciente $\{0 = \varepsilon_0 < \varepsilon_1 < \dots < \varepsilon_m\}$, tenemos la filtración de Vietoris-Rips

$$\text{VR}(N; \varepsilon_0) \subset \text{VR}(N; \varepsilon_1) \subset \dots \subset \text{VR}(N; \varepsilon_m).$$

Podemos denotar a $\text{VR}(N; \varepsilon_i)$ simplemente por VR_i cuando no exista riesgo de confusión a cerca de la nube de puntos o de la filtración. En [25] se pueden encontrar varios algoritmos para la construcción de esta estructura simplicial.

La filtración de Vietoris-Rips es una herramienta muy útil en las aplicaciones de topología algebraica para el análisis de datos. La manera de estudiar una nube de datos con el complejo de Vietoris-Rips (o alguna otra estructura de complejo simplicial) es a través de homología persistente.

Dada una secuencia de complejos simpliciales $\Delta_0 \subset \Delta_1 \subset \dots \subset \Delta_m$, para $i, j \in \{0, 1, \dots, m\}$ tales que $i \leq j$, el (i, j) -grupo de homología persistente p -dimensional $H_p^{i,j}$ de la filtración está definido como $\text{Im}(H_p^\Delta(\Delta_i) \rightarrow H_p^\Delta(\Delta_j))$, donde $H_*^\Delta(-)$ denota el funtor de homología simplicial. Ver [26] para un análisis más profundo.

Este complejo simplicial tiene muchos simplejos, sin embargo, una reducción apropiada o simplificación puede ser útil. Un trabajo en esta dirección se puede encontrar en [10]. Por otra parte, existen varias bibliotecas para calcular homología persistente; ver [22] para una lista extensa.

Una de estas bibliotecas es el software Ripser (cf. [1]), un software muy eficiente para calcular homología persistente del complejo de Vietoris-Rips. El siguiente lema proporciona un método alternativo para calcular la homología persistente de tal complejo simplicial, a través de reducciones \mathcal{J} -contraíbles mediante vértices.

Teorema 3.3.1. Sea $\text{VR}_0 \subset \text{VR}_1 \subset \dots \subset \text{VR}_m$ la filtración de Vietoris-Rips para alguna nube de puntos finita, y sea $H_p^{i,j}$ el subgrupo de (i, j) -homología persistente p -dimensional, asociada a la filtración. Si $G_i := \text{VR}_i^{(1)}$ es la gráfica correspondiente al 1-esqueleto del complejo simplicial VR_i y S_i es una sucesión de vértices \mathcal{J}_f -contraíbles de G_i para todo $1 \leq i \leq m$, entonces

$$H_p^{i,j} \cong \text{Im}(\iota_* : H_p(\mathcal{I}(G_i; S_i)) \rightarrow H_p(\mathcal{I}(G_j; S_j))). \quad (3.2)$$

Demostración. Por definición $H_p^\Delta(\text{VR}_i) \cong H_p(G_i)$ y del Teorema 3.2.2, tenemos el diagrama conmutativo

$$\begin{array}{ccc}
 H_p^\Delta(\text{VR}_i) \cong H_p(G_i) & \xrightarrow{\quad\quad\quad} & H_p^\Delta(\text{VR}_{i+1}) \cong H_p(G_{i+1}) \\
 I_*(S_i) \downarrow \cong & & I_*(S_j) \downarrow \cong \\
 H_p^\Delta(\Delta(I(G_i; S_i))) \cong H_p(I(G_i; S_i)) & \xrightarrow{\quad \iota_* \quad} & H_p^\Delta(\Delta(I(G_{i+1}; S_{i+1}))) \cong H_p(I(G_{i+1}; S_{i+1})).
 \end{array}$$

El isomorfismo (3.2) es una consecuencia de la functorialidad de la teoría de homología y el teorema de la persistencia equivalente [11, Sec. VII.2]. ■

Otra manera de calcular la homología persistente es a través de reducciones procedentes de la teoría de Morse discreta (cf. [12]). Este enfoque no es trivial porque la pareja libre removida debe preservar la estructura de filtración y la homología persistente; ver [21] para detalles.

En [4], se puede acceder al software Perseus, que realiza ciertas reducciones utilizando Teoría de Morse que preservan la homología en varias estructuras, incluida la simplicial.

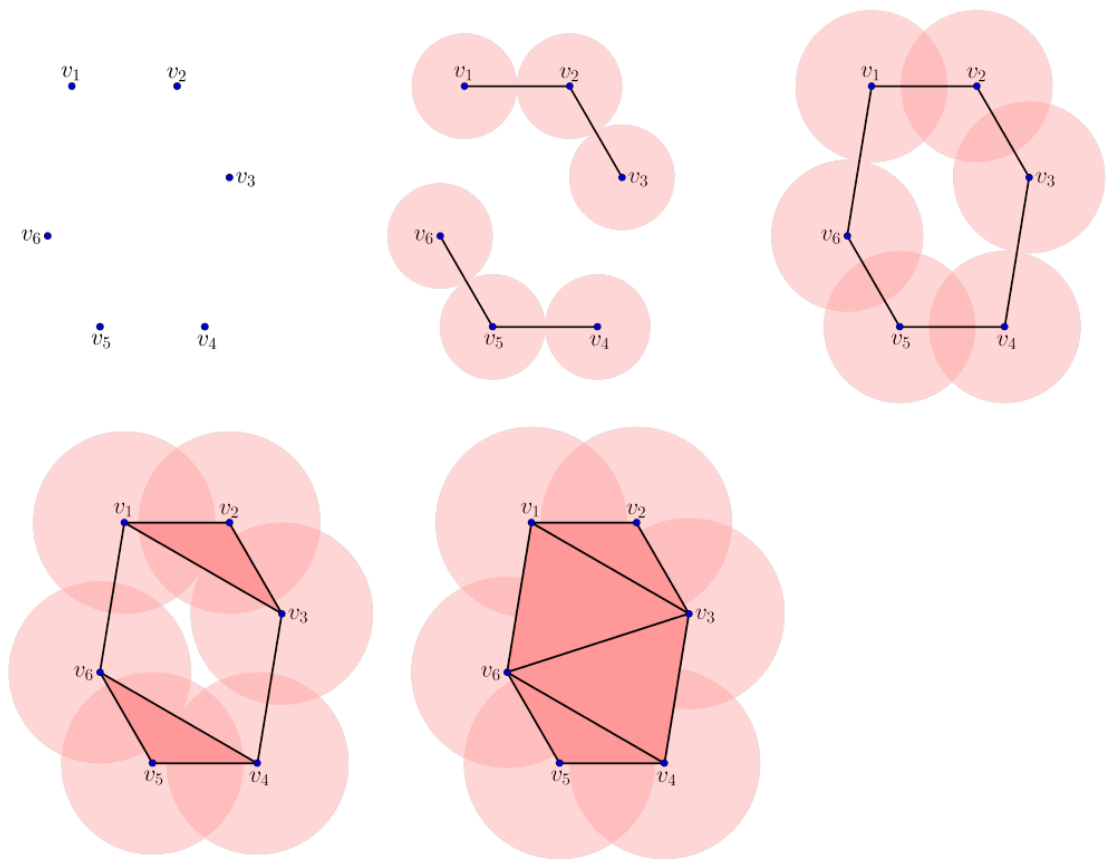
Nuestro enfoque basado en reducciones contraíbles no cuida la preservación de las filtraciones de las reducciones; sin embargo, los homomorfismos inducidos son en realidad la preservación de homología persistente como lo establece el Teorema 3.3.1.

Concluimos esta sección con un ejemplo de la homología persistente de el complejo de Vietoris-Rips de una nube de puntos en el plano.

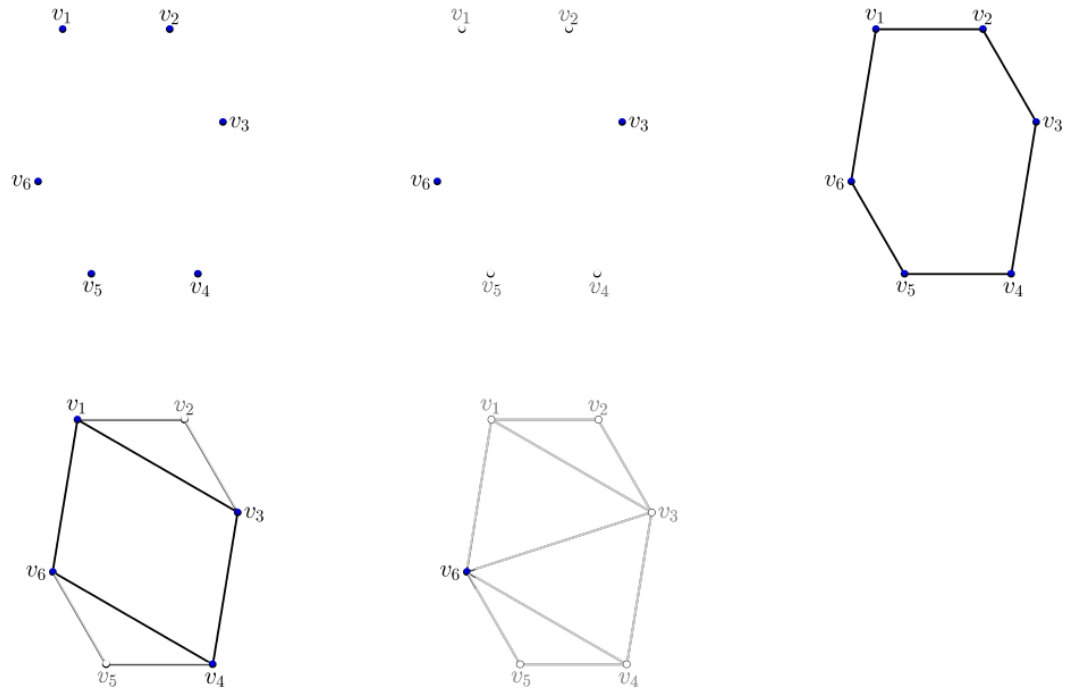
Ejemplo 3.3.2. Sea $N = \{v_1, \dots, v_6\} \subset \mathbb{R}^2$ una nube de puntos, y sea

$$\text{VR}(N; \varepsilon_1 = 0) \subset \text{VR}(N; \varepsilon_2 = 1,5) \subset \text{VR}(N; \varepsilon_3 = 2,1) \subset \text{VR}(N; \varepsilon_4 = 2,6) \subset \text{VR}(N; \varepsilon_5 = 2,7)$$

la filtración de Vietoris-Rips, como se muestra en la siguiente figura.

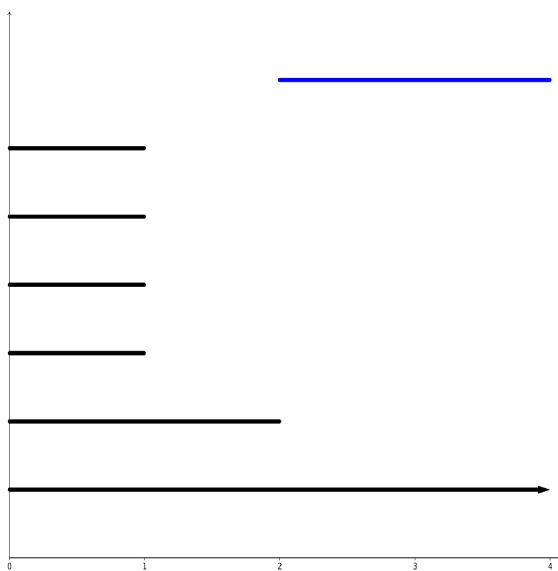


La siguiente figura muestra la secuencia de gráficas dadas por el algoritmo `vertex.contraible.reduction` sobre cada complejo simplicial $G_i := I(\text{VR}(\mathbb{N}; \varepsilon_i))$.



Claramente la secuencia $\{G_i\}$ no tiene una estructura de filtración. La siguiente tabla muestra los generadores de los grupos de p -homología (de cada gráfica reducida). A la derecha se aprecia el código de barras para la homología persistente (los cálculos se realizaron con coeficientes en el campo $\mathbb{Z}/2\mathbb{Z}$).

	$H_p(G_1)$	$H_p(G_2)$	$H_p(G_3)$	$H_p(G_4)$	$H_p(G_5)$
$p = 0$	$[v_1]$	$[v_3]$	$[v_1]$	$[v_1]$	$[v_6]$
	$[v_2]$				
	$[v_3]$				
	$[v_4]$	$[v_6]$			
	$[v_5]$				
	$[v_6]$				
$p = 1$	0	0	α	β	0



estamos denotando $\alpha = [v_1v_2] + [v_2v_3] + [v_3v_4] + [v_4v_5] + [v_5v_6] + [v_1v_6]$ y $\beta = [v_1v_3] + [v_3v_4] + [v_4v_6] + [v_1v_6]$.

En el repositorio en [3], se encuentra material visual a cerca de la construcción del complejo de Vietoris-Rips (como gráfica), seguido de la correspondiente estructura reducida contraíble. Los experimentos muestran un comportamiento similar al ejemplo anterior.

Capítulo 4

Conclusiones y trabajo a futuro

4.1. Resultados obtenidos

- Se ha elaborado un código propio, eficiente para resolver el problema de isomorfismo entre gráficas simples pequeñas.
- Se ha probado la existencia de una subfamilia propia de la familia \mathcal{I} -contraíble, a saber \mathcal{I}_f -contraíbles.
- Se han refutado al menos tres afirmaciones realizadas en [16], respecto a propiedades de las gráficas \mathcal{I} -contraíbles, dejando sin prueba válida los resultados que de ellos dependen.
- Se ha probado que $\mathcal{I}_f \subset \mathcal{C}$.
- Se tiene evidencia que $\mathcal{I}_f = \mathcal{I} = \mathcal{C}$, para un subconjunto amplio de gráficas limitado a nueve vértices.
- Se ha implementado reducciones en el complejo de Vietoris-Rips utilizando \mathcal{I}_f -contracciones por vértices. Lo cual proporciona una importante alternativa para el cálculo de la homología persistente de dicha estructura simplicial.
- Se ha demostrado que la gráfica *dunce hat*, es \mathcal{I} -contraíble.
- Se cuenta con código reutilizable para otros propósitos sobre gráficas simples.

4.2. Trabajo a futuro

- Se requiere mejorar los algoritmos a fin de aumentar los alcances de los resultados. Entre otros aspectos se requiere mejorar la estructura de datos.
- Es necesario probar o refutar que el orden de eliminación de vértices no importa, en la familia $\langle K(1), I2 \rangle$, es decir la familia que se obtiene apartir de $k(1)$ utilizando sólo la operación de pegado de vértice.
- Es necesario probar o refutar que $\mathcal{C} \subseteq \mathcal{I}_f$.
- Diseñar software para el cálculo de homología persistente del complejo de Vietoris-Rips usando \mathcal{I} -contracciones.

- Probar o refutar que $\mathcal{I} = \mathfrak{S}\mathcal{C}$.

Apéndices

Apéndice A

Ejemplos de gráficas conexas y fuertemente \mathfrak{J} -contraíbles

A.1. Primeras 50 gráficas conexas

En la siguiente página se muestran las gráficas conexas obtenidas en la Sección 2.1, ahora están agrupadas en subconjuntos por cantidad de vértices y aristas $[|V|, |E|]$.

En páginas más adelante se muestra la obtención de cada una de las gráficas al aplicar las operaciones dadas en 2.1.

Op1 corresponde a agregar vértice y arista, se indica con $G_{old} + v_j + \{v_i, v_j\}$.

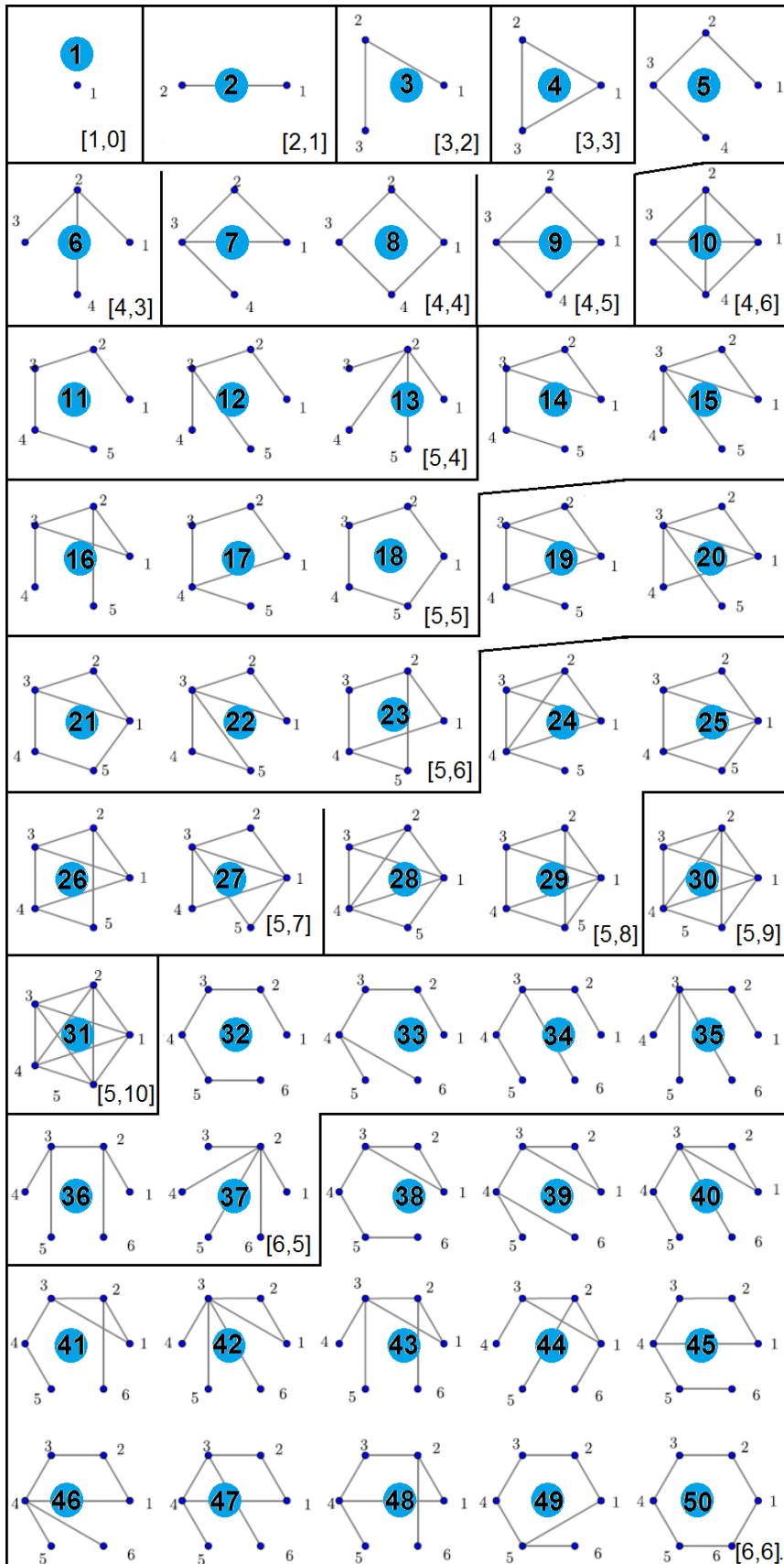
Op2 agregar sólo arista, se indica con $G_{old} + \{v_i, v_j\}$.

En caso de que la gráfica resultante se isomorfa a una gráfica ya obtenida, se indica a cuál es isomorfa. En otro caso, se almacena asignándole un índice.¹

La primera gráfica es $G_1 = K(1)$ y la última de la lista es el ciclo de seis aristas.

¹Se invita al lector a verificar la construcción.

Apéndice A



[1, 0] $G_1 = \{v_1\}$ (el inicio) -----	$G_{16} \leftarrow G_7 + v_5 + \{2, 5\}$ $G_7 + v_5 + \{1, 5\} \cong G_{16}$
[2, 1] $G_2 \leftarrow G_1 + v_2 + \{1, 2\}$ -----	$G_{17} \leftarrow G_8 + v_5 + \{4, 5\}$ $G_8 + v_5 + \{3, 5\} \cong G_{17}$ $G_8 + v_5 + \{2, 5\} \cong G_{17}$ $G_8 + v_5 + \{1, 5\} \cong G_{17}$
[3, 2] $G_3 \leftarrow G_2 + v_3 + \{2, 3\}$ $G_2 + v_3 + \{1, 3\} \cong G_3$ -----	$G_{11} + \{1, 3\} \cong G_{14}$ $G_{11} + \{1, 4\} \cong G_{17}$ $G_{18} \leftarrow G_{11} + \{1, 5\}$ $G_{11} + \{2, 4\} \cong G_{16}$
[3, 3] $G_4 \leftarrow G_3 + \{1, 3\}$ -----	$G_{11} + \{2, 5\} \cong G_{17}$ $G_{11} + \{3, 5\} \cong G_{14}$ $G_{12} + \{1, 3\} \cong G_{14}$
[4, 3] $G_5 \leftarrow G_3 + v_4 + \{3, 4\}$ $G_6 \leftarrow G_3 + v_4 + \{2, 4\}$ $G_3 + v_4 + \{1, 4\} \cong G_5$ -----	$G_{12} + \{1, 4\} \cong G_{17}$ $G_{12} + \{1, 5\} \cong G_{17}$ $G_{12} + \{2, 4\} \cong G_{16}$ $G_{12} + \{2, 5\} \cong G_{16}$ $G_{12} + \{4, 5\} \cong G_{14}$
[4, 4] $G_7 \leftarrow G_4 + v_4 + \{3, 4\}$ $G_4 + v_4 + \{2, 4\} \cong G_7$ $G_4 + v_4 + \{1, 4\} \cong G_7$ $G_5 + \{1, 3\} \cong G_7$ $G_8 \leftarrow G_5 + \{1, 4\}$ $G_5 + \{2, 4\} \cong G_7$ -----	$G_{13} + \{1, 3\} \cong G_{15}$ $G_{13} + \{1, 4\} \cong G_{15}$ $G_{13} + \{1, 5\} \cong G_{15}$ $G_{13} + \{3, 4\} \cong G_{15}$ $G_{13} + \{3, 5\} \cong G_{15}$ $G_{13} + \{4, 5\} \cong G_{15}$ -----
[4, 5] $G_9 \leftarrow G_7 + \{1, 4\}$ $G_7 + \{2, 4\} \cong G_9$ -----	[5, 6] $G_{19} \leftarrow G_9 + v_5 + \{4, 5\}$ $G_{20} \leftarrow G_9 + v_5 + \{3, 5\}$ $G_9 + v_5 + \{2, 5\} \cong G_{19}$ $G_9 + v_5 + \{1, 5\} \cong G_{20}$ $G_{14} + \{1, 4\} \cong G_{19}$
[4, 6] $G_{10} \leftarrow G_9 + \{2, 4\}$ -----	$G_{21} \leftarrow G_{14} + \{1, 5\}$ $G_{14} + \{2, 4\} \cong G_{19}$ $G_{14} + \{2, 5\} \cong G_{21}$
[5, 4] $G_{11} \leftarrow G_5 + v_5 + \{4, 5\}$ $G_{12} \leftarrow G_5 + v_5 + \{3, 5\}$ $G_5 + v_5 + \{2, 5\} \cong G_{12}$ $G_5 + v_5 + \{1, 5\} \cong G_{11}$ $G_6 + v_5 + \{4, 5\} \cong G_{12}$ $G_6 + v_5 + \{3, 5\} \cong G_{12}$ $G_{13} \leftarrow G_6 + v_5 + \{2, 5\}$ $G_6 + v_5 + \{1, 5\} \cong G_{12}$ -----	$G_{22} \leftarrow G_{14} + \{3, 5\}$ $G_{15} + \{4, 5\} \cong G_{21}$ $G_{16} + \{1, 4\} \cong G_{19}$ $G_{16} + \{1, 5\} \cong G_{19}$ $G_{16} + \{2, 4\} \cong G_{20}$ $G_{16} + \{3, 5\} \cong G_{20}$ $G_{16} + \{4, 5\} \cong G_{21}$ $G_{17} + \{1, 3\} \cong G_{19}$
[5, 5] $G_{14} \leftarrow G_7 + v_5 + \{4, 5\}$ $G_{15} \leftarrow G_7 + v_5 + \{3, 5\}$	$G_{23} \leftarrow G_{17} + \{2, 5\} \cong G_{21}$ $G_{17} + \{3, 5\} \cong G_{21}$ -----

[5, 7]	[6, 5]
$G_{24} \leftarrow G_{10} + v_5 + \{4, 5\}$	$G_{32} \leftarrow G_{11} + v_6 + \{5, 6\}$
$G_{10} + v_5 + \{3, 5\} \cong G_{24}$	$G_{33} \leftarrow G_{11} + v_6 + \{4, 6\}$
$G_{10} + v_5 + \{2, 5\} \cong G_{24}$	$G_{34} \leftarrow G_{11} + v_6 + \{3, 6\}$
$G_{10} + v_5 + \{1, 5\} \cong G_{24}$	$G_{11} + v_6 + \{2, 6\} \cong G_{33}$
$G_{25} \leftarrow G_{19} + \{1, 5\}$	$G_{11} + v_6 + \{1, 6\} \cong G_{32}$
$G_{19} + \{2, 4\} \cong G_{24}$	$G_{12} + v_6 + \{5, 6\} \cong G_{34}$
$G_{26} \leftarrow G_{19} + \{2, 5\}$	$G_{12} + v_6 + \{4, 6\} \cong G_{34}$
$G_{19} + \{3, 5\} \cong G_{25}$	$G_{35} \leftarrow G_{12} + v_6 + \{3, 6\}$
$G_{27} \leftarrow G_{20} + \{2, 5\}$	$G_{36} \leftarrow G_{12} + v_6 + \{2, 6\}$
$G_{20} + \{4, 5\} \cong G_{25}$	$G_{12} + v_6 + \{1, 6\} \cong G_{33}$
$G_{21} + \{1, 4\} \cong G_{25}$	$G_{13} + v_6 + \{5, 6\} \cong G_{35}$
$G_{21} + \{2, 4\} \cong G_{26}$	$G_{13} + v_6 + \{4, 6\} \cong G_{35}$
$G_{21} + \{2, 5\} \cong G_{26}$	$G_{13} + v_6 + \{3, 6\} \cong G_{35}$
$G_{21} + \{3, 5\} \cong G_{25}$	$G_{37} \leftarrow G_{13} + v_6 + \{2, 6\}$
$G_{22} + \{1, 4\} \cong G_{25}$	$G_{13} + v_6 + \{1, 6\} \cong G_{35}$
$G_{22} + \{1, 5\} \cong G_{25}$	-----
$G_{22} + \{2, 4\} \cong G_{25}$	[6, 6]
$G_{22} + \{2, 5\} \cong G_{25}$	$G_{38} \leftarrow G_{14} + v_6 + \{5, 6\}$
$G_{23} + \{2, 4\} \cong G_{27}$	$G_{39} \leftarrow G_{14} + v_6 + \{4, 6\}$
$G_{22} + \{3, 5\} \cong G_{26}$	$G_{40} \leftarrow G_{14} + v_6 + \{3, 6\}$
-----	$aG_{41} \leftarrow G_{14} + v_6 + \{2, 6\}$
[5, 8]	$G_{14} + v_6 + \{1, 6\} \cong G_{41}$
$G_{28} \leftarrow G_{24} + \{1, 5\}$	$G_{15} + v_6 + \{5, 6\} \cong G_{40}$
$G_{24} + \{2, 5\} \cong G_{28}$	$G_{15} + v_6 + \{4, 6\} \cong G_{40}$
$G_{24} + \{3, 5\} \cong G_{28}$	$G_{42} \leftarrow G_{15} + v_6 + \{3, 6\}$
$G_{25} + \{2, 4\} \cong G_{28}$	$G_{43} \leftarrow G_{15} + v_6 + \{2, 6\}$
$G_{29} \leftarrow G_{25} + \{2, 5\}$	$G_{15} + v_6 + \{1, 6\} \cong G_{43}$
$G_{25} + \{3, 5\} \cong G_{28}$	$G_{16} + v_6 + \{5, 6\} \cong G_{41}$
$G_{26} + \{1, 5\} \cong G_{29}$	$G_{16} + v_6 + \{4, 6\} \cong G_{41}$
$G_{26} + \{2, 4\} \cong G_{28}$	$G_{16} + v_6 + \{3, 6\} \cong G_{43}$
$G_{26} + \{3, 5\} \cong G_{29}$	$G_{16} + v_6 + \{2, 6\} \cong G_{43}$
$G_{27} + \{2, 4\} \cong G_{28}$	$G_{44} \leftarrow G_{16} + v_6 + \{1, 6\}$
$G_{27} + \{2, 5\} \cong G_{28}$	$G_{45} \leftarrow G_{17} + v_6 + \{5, 6\}$
$G_{27} + \{4, 5\} \cong G_{28}$	$G_{46} \leftarrow G_{17} + v_6 + \{4, 6\}$
-----	$G_{47} \leftarrow G_{17} + v_6 + \{3, 6\}$
[5, 9]	$G_{48} \leftarrow G_{17} + v_6 + \{2, 6\}$
$G_{30} \leftarrow G_{28} + \{2, 5\}$	$G_{17} + v_6 + \{1, 6\} \cong G_{47}$
$G_{28} + \{3, 5\} \cong G_{30}$	$G_{49} \leftarrow G_{18} + v_6 + \{1, 6\}$
$G_{29} + \{2, 4\} \cong G_{30}$	$G_{18} + v_6 + \{2, 6\} \cong G_{49}$
$G_{29} + \{3, 5\} \cong G_{30}$	$G_{18} + v_6 + \{3, 6\} \cong G_{49}$
-----	$G_{18} + v_6 + \{2, 6\} \cong G_{49}$
[5, 10]	$G_{18} + v_6 + \{1, 6\} \cong G_{49}$
$G_{31} \leftarrow G_{28} + \{3, 5\}$	$G_{32} + \{1, 3\} \cong G_{38}$
	$G_{32} + \{1, 4\} \cong G_{45}$
	$G_{32} + \{1, 5\} \cong G_{49}$
	$G_{50} \leftarrow G_{32} + \{1, 6\}$

Observaciones. A.1.1.

- Para detectar pares de gráficas isomorfas se utilizó la técnica de etiquetado canónico, y a cada gráfica se le asignó su número máximo.
- Cada gráfica es representante de su clase de isomorfismo, en todos los casos fue la primera de su clase en ser generada, es decir, no fue sustituida por alguna otra de su clase.
- Una vez generadas las gráficas se sugiere ordenar por su número máximo dentro de cada subconjunto $[|V|, |E|]$.
- La memoria de cálculo sugiere mejoras menores a la presente propuesta de generación de gráficas conexas, en el sentido de ser visualmente más apropiadas.

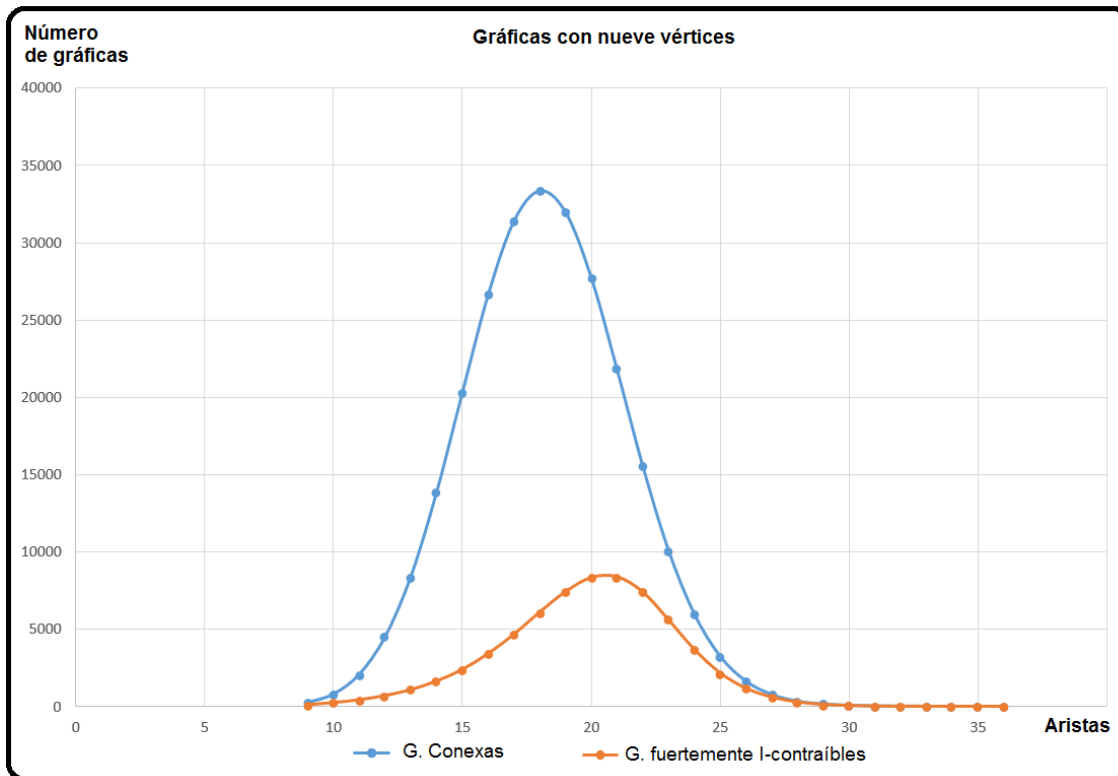
A.2. Primeras gráficas fuertemente \mathcal{I} -contraíbles

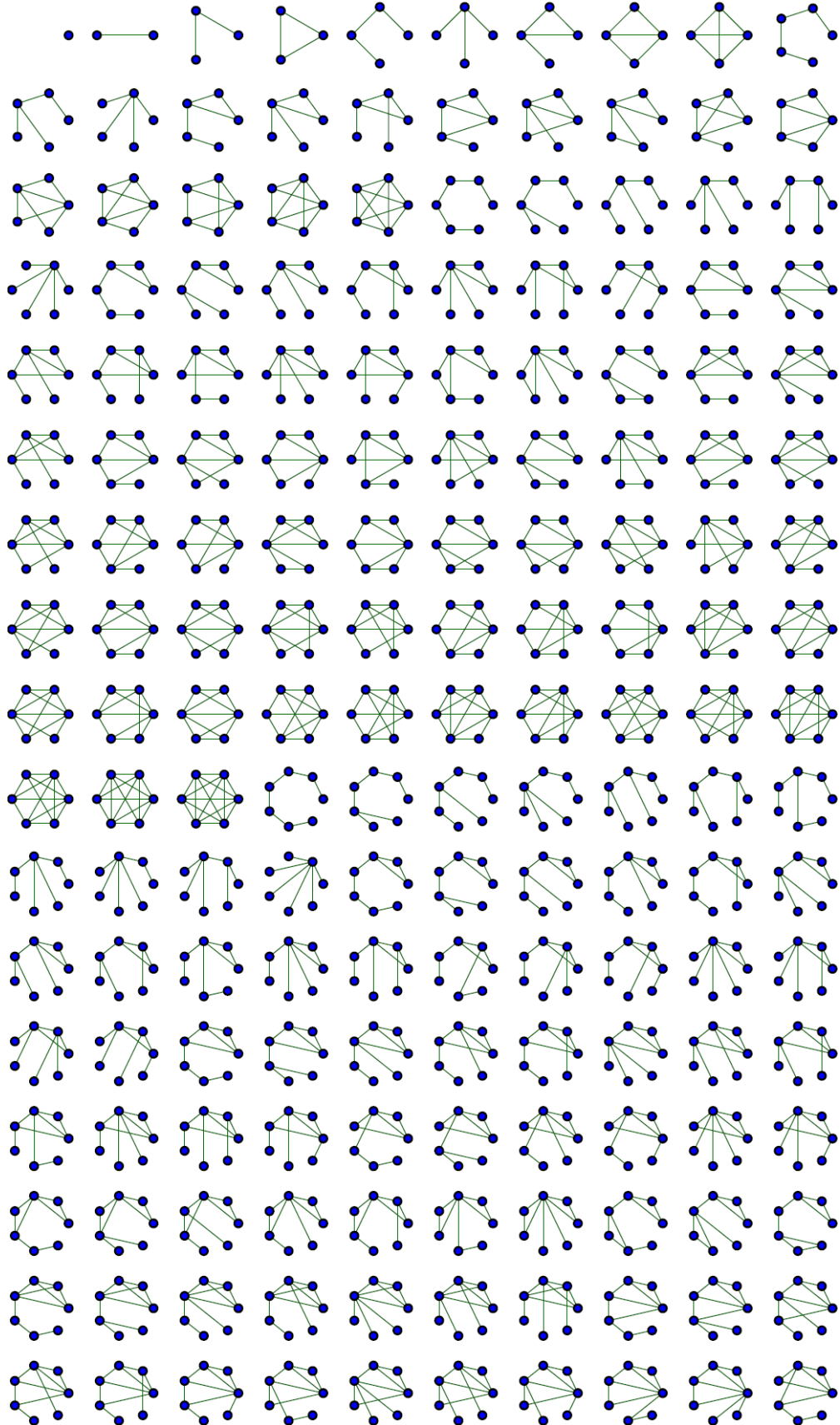
En las siguientes dos páginas se muestra un subconjunto de 340 gráficas fuertemente \mathcal{I} -contraíbles, fueron obtenidas aplicando el Algoritmo 3. Verificándose que coinciden con gráficas colapsables.

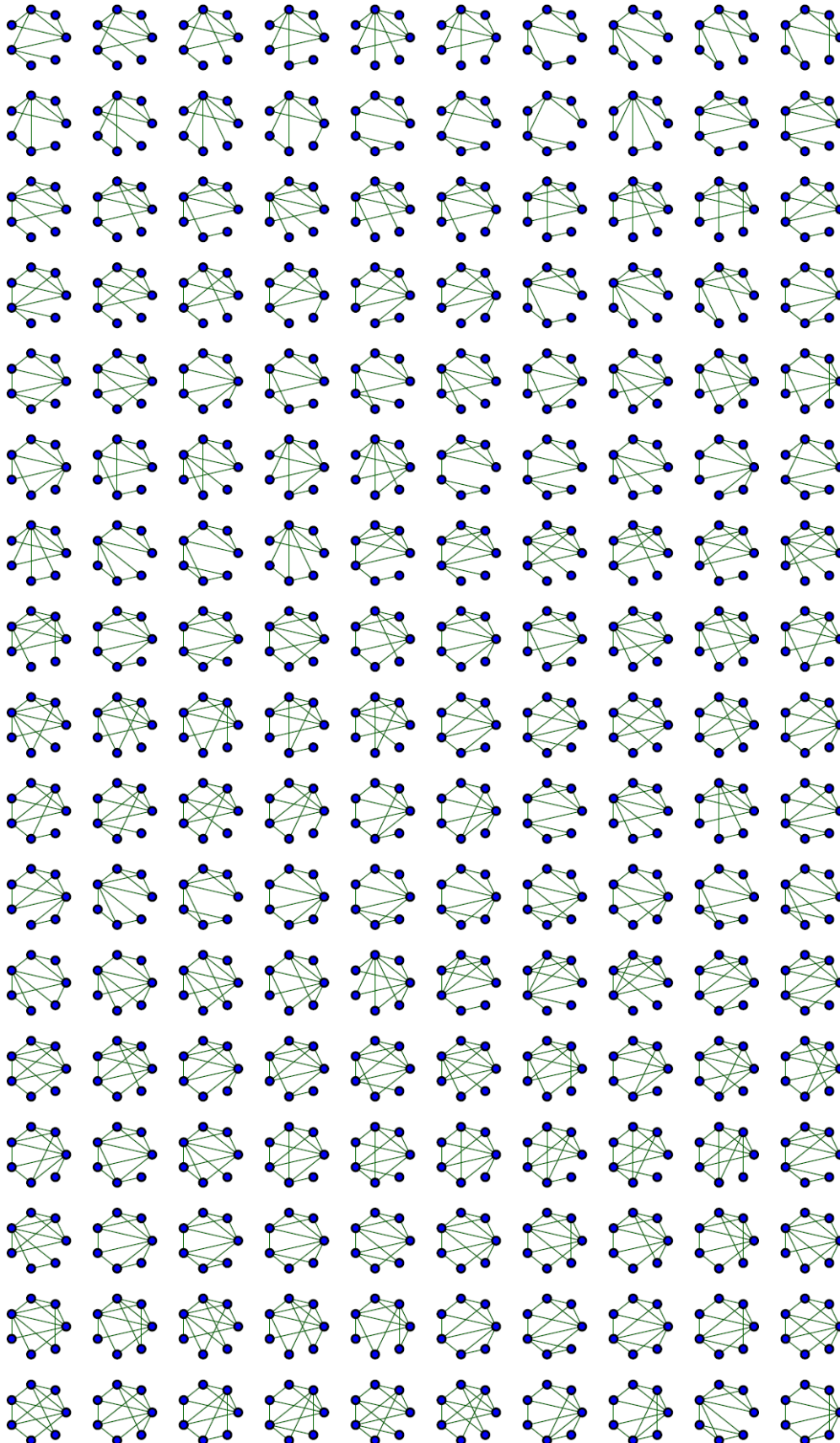
Se agrupan por número de vértice y de número de aristas, no están precisamente ordenadas por número máximo, aunque la mayoría sigue ese orden.

A fin de no saturar el dibujo se omiten las etiquetas de los vértices. No aparecen en su totalidad las gráficas de siete vértices \mathcal{I}_f -contraíbles. Al igual que las gráficas conexas ya mostradas, estas gráficas también parecen ser visualmente más comprensibles comparadas con otras colecciones, ver [2], aunque estas últimas sean sólo gráficas conexas.

La siguiente figura muestra la cantidad de gráficas conexas de nueve vértices que hay de cada cantidad de aristas, junto con las gráficas \mathcal{I}_f -contraíbles.



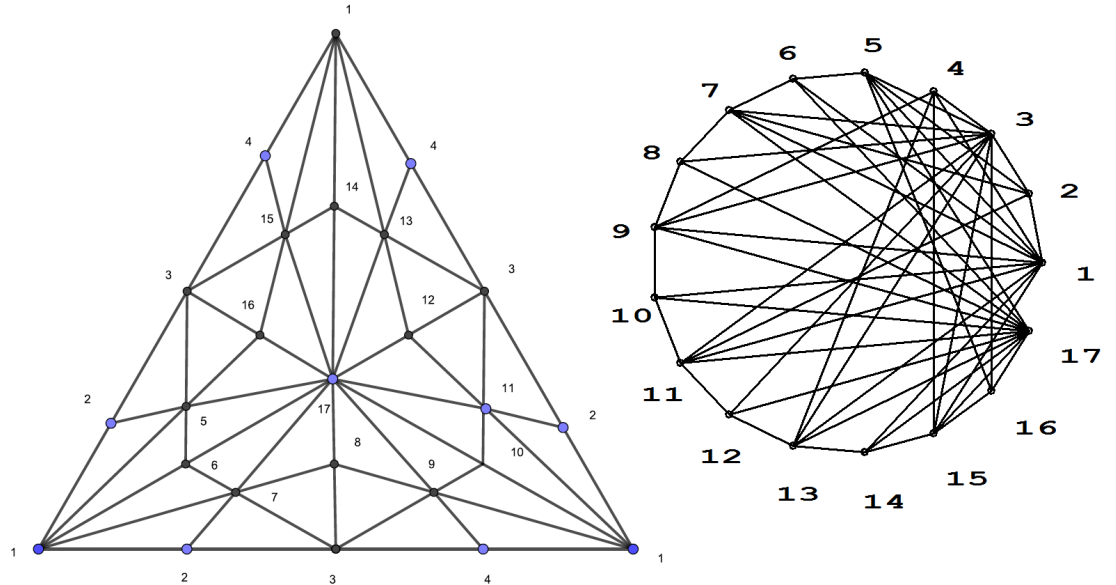




Apéndice B

Transformación del *dunce hat* a $K(17)$

Representaciones geométricas de la gráfica *dunce hat*; estándar y con vértices sobre S_1 .



Matriz de adyacencia:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	0	1	0	1	1	1	1	0	1	1	1	0	1	1	1	0	0
2	1	0	1	0	1	0	1	0	0	0	1	0	0	0	0	0	0
3	0	1	0	1	1	0	1	1	1	0	1	1	1	0	1	1	0
4	1	0	1	0	0	0	0	0	1	0	0	0	1	0	1	0	0
5	1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	1	1
6	1	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	1
7	1	1	1	0	0	1	0	1	0	0	0	0	0	0	0	0	1
8	0	0	1	0	0	0	1	0	1	0	0	0	0	0	0	0	1
9	1	0	1	1	0	0	0	1	0	1	0	0	0	0	0	0	1
10	1	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	1
11	1	1	1	0	0	0	0	0	0	1	0	1	0	0	0	0	1
12	0	0	1	0	0	0	0	0	0	0	1	0	1	0	0	0	1
13	1	0	1	1	0	0	0	0	0	0	0	1	0	1	0	0	1
14	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1
15	1	0	1	1	0	0	0	0	0	0	0	0	0	1	0	1	1
16	0	0	1	0	1	0	0	0	0	0	0	0	0	0	1	0	1
17	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0

A continuación se muestra la evidencia gráfica que de que dunce hat se puede transformar en una gráfica completa agregando aristas \mathcal{J}_f -contraíbles, para detectar vecindades \mathcal{J}_f -contraíbles se utilizó el Algoritmo 3.

Para cada arista $\{i, j\}$ a agregar, el vértice i , las aristas que inciden en él y los círculos que encierran a los vértices de $N(i)$ son de color verde, para el vértice j se utiliza el color morado, así los vértices de $N(\{i, j\})$ se muestran doblemente circulados.

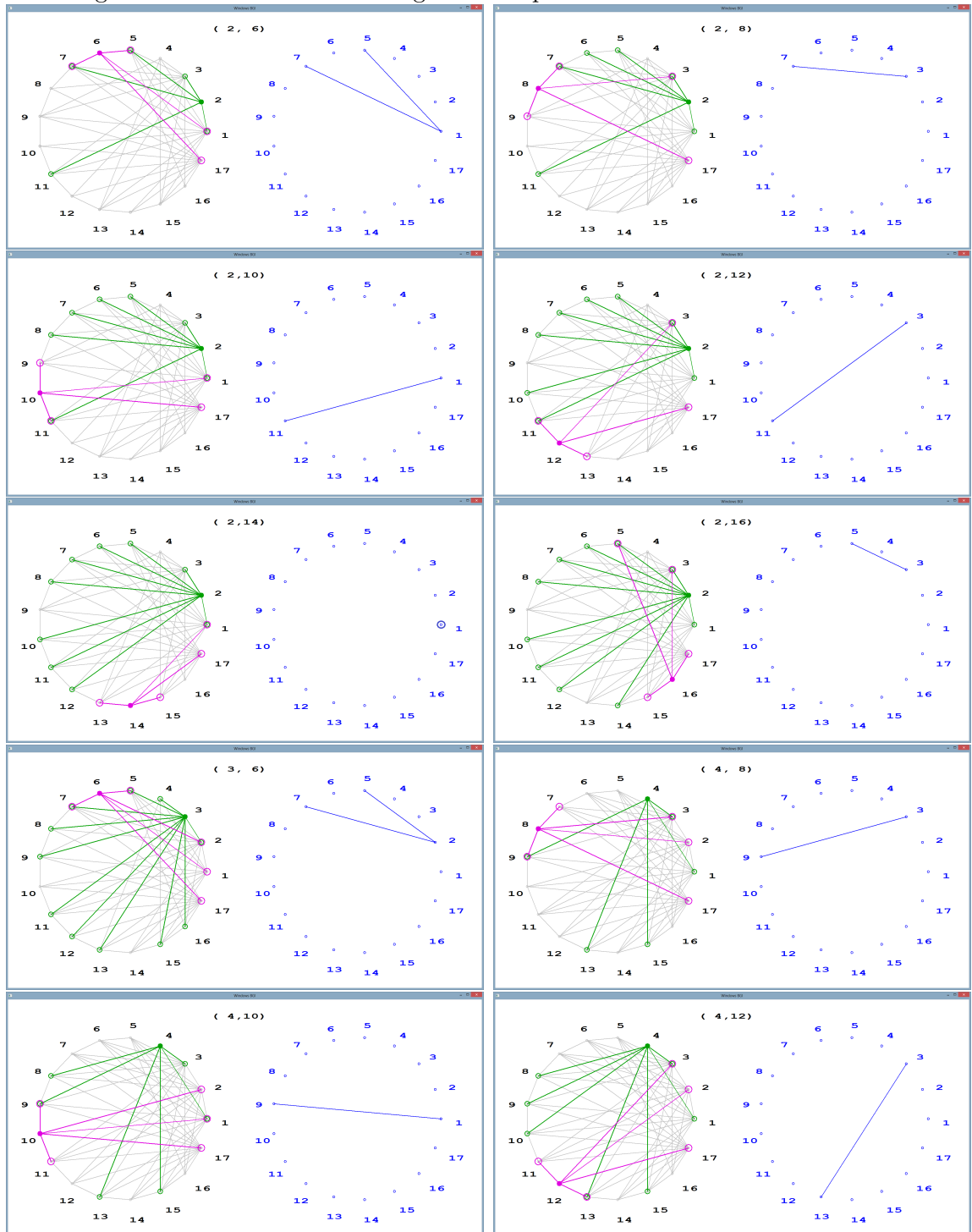
Las gráficas $N(\{i, j\})$ se muestran al lado derecho en color azul, para verificar en el dibujo que efectivamente $N(\{i, j\})$ es \mathcal{J}_f -contraíble.

Enseguida se muestra una secuencia de aristas que pueden ser agregadas para transformar la gráfica dunce hat a una gráfica completa.

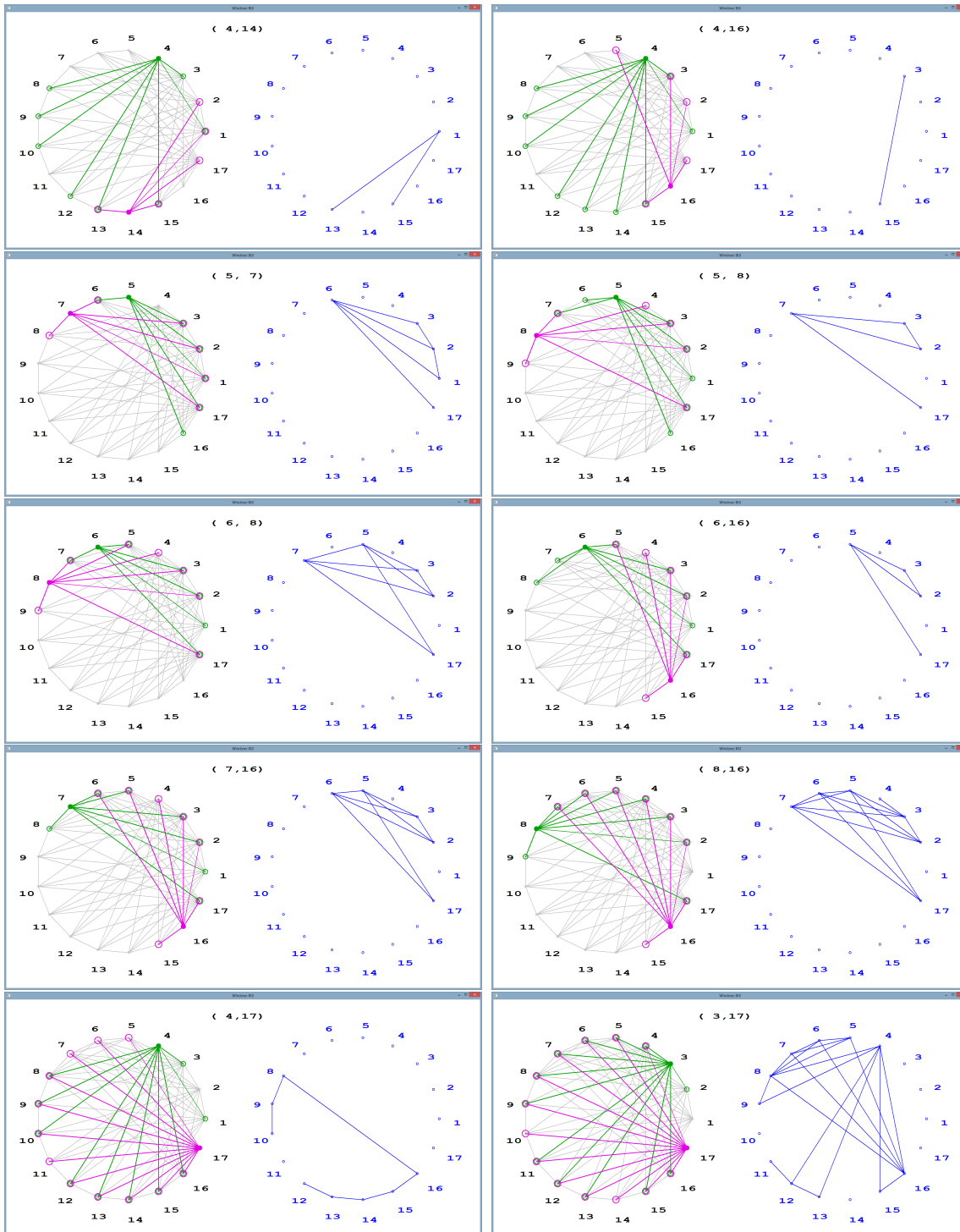
- {2, 6} {2, 8} {2, 10} {2, 12} {2, 14} {2, 16} {3, 6} {4, 8} {4, 10} {4, 12}
- {4, 14} {4, 16} {5, 7} {5, 8} {6, 8} {6, 16} {7, 16} {8, 16} {4, 17} {3, 17}
- {3, 10} {1, 3} {1, 8} {1, 12} {1, 16} {1, 17} {2, 4} {2, 9} {2, 13} {2, 15}
- {2, 17} {3, 14} {4, 5} {4, 6} {4, 7} {4, 11} {5, 9} {5, 10} {5, 11} {5, 12}
- {5, 13} {5, 14} {5, 15} {6, 9} {6, 10} {6, 11} {6, 12} {6, 13} {6, 14} {6, 15}
- {7, 9} {7, 10} {7, 11} {7, 12} {7, 13} {7, 14} {7, 15} {8, 10} {8, 11} {8, 12}
- {8, 13} {8, 14} {8, 15} {9, 11} {9, 12} {9, 13} {9, 14} {9, 15} {9, 16} {10, 12}
- {10, 13} {10, 14} {10, 15} {10, 16} {11, 13} {11, 14} {11, 15} {11, 16} {12, 14} {12, 15}
- {12, 16} {13, 15} {13, 16} {14, 16}

Al agregar la arista $\{1, 17\}$ se obtiene un cono con vértice en 1, ver la Figura B.1. Este cono puede ser \mathcal{J}_f -contraído a $K(1)$, así la gráfica dunce hat pertenece a la familia de gráficas \mathcal{J} -contraíbles.

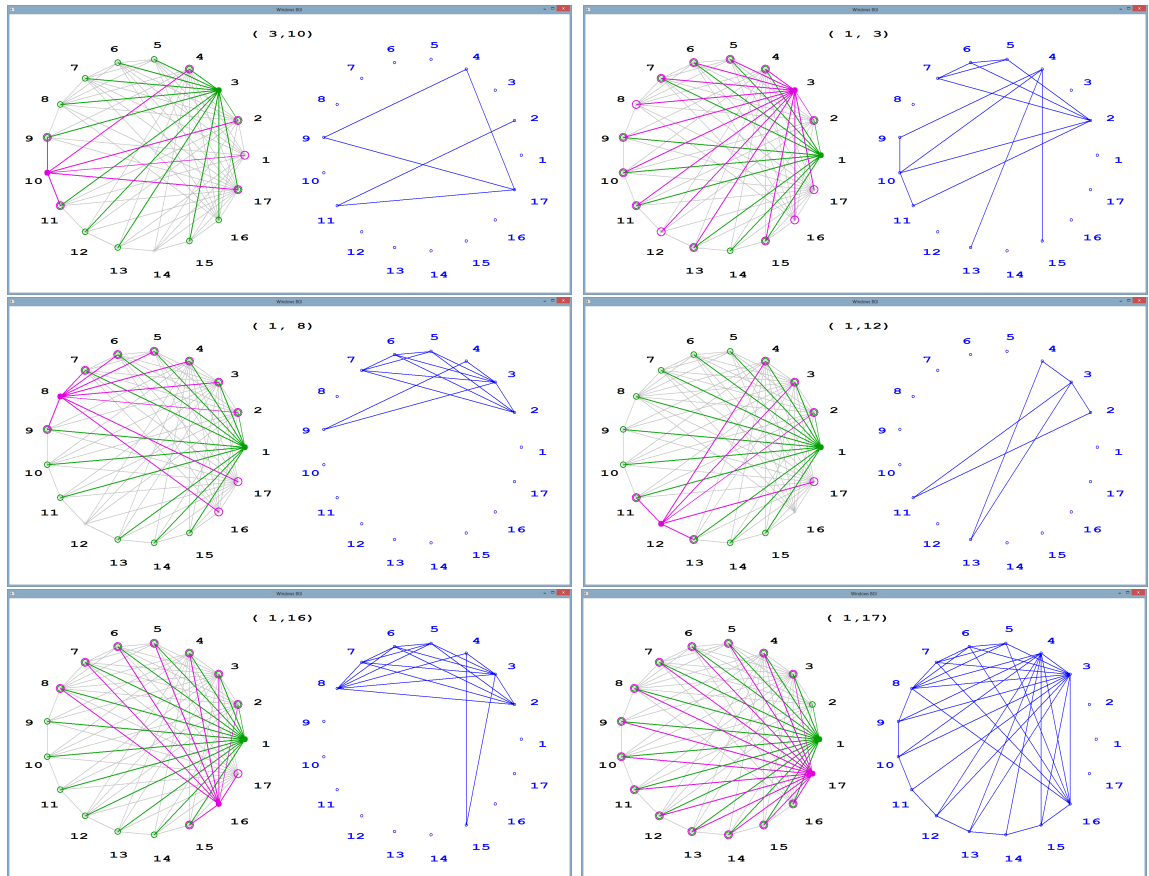
Enseguida se muestra la memoria gráfica del proceso.



Apéndice B



Transformación del *dunce hat* a $K(17)$



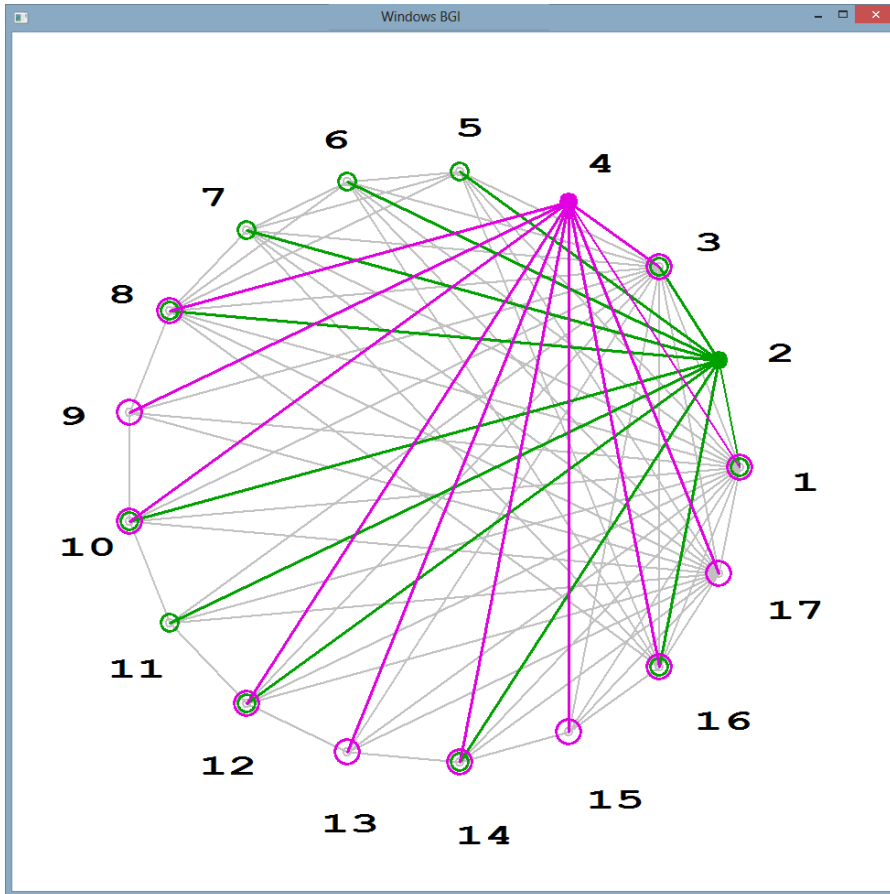


Figura B.1: Cono con vértice en 1; fue obtenido en el proceso de llevar a la gráfica *dunce hat* a $K(17)$.

Bibliografía

- [1] Bauer U (2016) Ripser. <https://github.com/Ripser/ripser>.
- [2] Gráficas obtenidas de la base de datos de mckay: <http://www.cadaeic.net/graphpics.htm>.
- [3] Repository *Geometric and Combinatorial Structures*. Research group at University of Sonora.
- [4] Vidit Nanda. Perseus, the Persistent Homology Software. <http://www.sas.upenn.edu/~vnanda/perseus>.
- [5] R. Boulet, E. Fieux, and B. Jouve. Simplicial simple-homotopy of flag complexes in terms of graphs. *European Journal of Combinatorics*, 31(1):161 – 176, 2010.
- [6] N. Boutry, T. Géraud, and L. Najman. A Tutorial on Well-Composedness. *Journal of Mathematical Imaging and Vision*, 60(3):443–478, Mar 2018.
- [7] G. Carlsson. Topology and data. *Bull Am Math Soc*, 46, 2009.
- [8] B. Chen, S.-T. Yau, and Y.-N. Yeh. Graph homotopy and graham homotopy. *Discrete Mathematics*, 241(1):153 – 170, 2001. Selected Papers in honor of Helge Tverberg.
- [9] A. Dharwadker and J.-T. Tevet. *The Graph Isomorphism Algorithm: Graph Isomorphism is in P*. CreateSpace, Paramount, CA, 2011.
- [10] P. Dotko and H. Wagner. Simplification of complexes of persistent homology computations. *Homology Homotopy Appl.*, 16(1):49–63, 2014.
- [11] H. Edelsbrunner and J. Harer. *Computational Topology: An Introduction*. Applied Mathematics. American Mathematical Society, 2010.
- [12] R. Forman. Morse theory for cell complexes. *Advances in Mathematics*, 134(1):90 – 145, 1998.
- [13] S.-E. Han. Contractibility and fixed point property: the case of Khalimsky topological spaces. *Fixed Point Theory and Applications*, 2016(1):75, Jun 2016.
- [14] A. Hatcher. *Algebraic Topology*. Algebraic Topology. Cambridge University Press, 2002.
- [15] A. V. Ivashchenko. Contractible transformations do not change the homology groups of graphs. *Discrete Mathematics*, 126(1):159 – 170, 1994.

-
- [16] A. V. Ivashchenko. Some properties of contractible transformations on graphs. *Discrete Mathematics*, 133(1):139 – 145, 1994.
- [17] M. Kahle. Topology of random simplicial complexes: a survey, 2014.
- [18] F. Larrión, M. A. Pizaña, and R. Villarroel-Flores. Iterated clique graphs and bordered compact surfaces. *Discrete Mathematics*, 313(4):508 – 516, 2013.
- [19] B. D. McKay. Database of graphs: <http://cs.anu.edu.au/~bdm/data/graphs.html>.
- [20] B. D. McKay and A. Piperno. Practical graph isomorphism, ii. *Journal of Symbolic Computation*, 60:94 – 112, 2014.
- [21] K. Mischaikow and V. Nanda. Morse theory for filtrations and efficient computation of persistent homology. *Discrete Comput. Geom.*, 50(2):330–353, Sept. 2013.
- [22] N. Otter, M. A. Porter, U. Tillmann, P. Grindrod, and H. A. Harrington. A roadmap for the computation of persistent homology. *EPJ Data Science*, 6(1):17, Aug 2017.
- [23] A. Patania, F. Vaccarino, and G. Petri. Topological analysis of data. *EPJ Data Science*, 6(1):7, Jun 2017.
- [24] E. Prisner. Convergence of iterated clique graphs. *Discrete Mathematics*, 103(2):199 – 207, 1992.
- [25] A. Zomorodian. Technical section: fast construction of the Vietoris–Rips complex. *Comput Graph*, 34, 2010.
- [26] A. Zomorodian and G. Carlsson. Computing persistent homology. *Discrete Comput Geom*, 33, 2005.