Índice

Aplicaciones de Coloración de Grafos

Penadillo Lazares, Wenses Johan
20182174D
wpenadillol@uni.pe

Raymundo Muñoa, Abrahan
20180562G
araymundom@uni.pe

Ramírez Mendoza, Jesús Daniel
20184115E
jesus.ramirez.m@uni.pe

1. Coloración de Grafos

En Teoría de grafos, la **coloración de grafos** es un caso especial de etiquetado de grafos; es una asignación de etiquetas llamadas colores a elementos del grafo. De manera simple, una coloración de los vértices de un grafo tal que ningún vértice adyacente comparta el mismo color es llamado vértice coloración. Similarmente, una arista coloración asigna colores a cada arista tal que aristas adyacentes no compartan el mismo color, y una coloración de caras de un grafo plano a la asignación de un color a cada cara o región tal que caras que compartan una frontera común tengan colores diferentes. El vértice coloración es el punto de inicio de la coloración, y los otros problemas de coloreo pueden ser transformados a una versión con vértices.

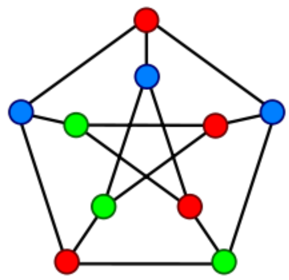


Figura 1

1.1. Vértice Coloración

La **vértice coloración** (o simplemente coloración) es la asignación de los vértices de un grafo con colores tal que dos vértices que compartan la misma arista tengan colores diferentes. Una coloración que usa a lo más k colores se llama k-coloración (propia). El menor número de colores necesarios para colorear un grafo G se llama **número cromático** y se denota como $\chi(G)$. Un grafo que puede ser asignada una k-coloración (propia) es k-coloreable y es k-cromático si su número cromático es exactamente k. Un subconjunto de vértices asignados con el mismo color se llama una clase de color. Cada clase forma un conjunto independiente. Esto es, una k-coloración es lo mismo que una partición del conjunto de vértices en k conjuntos independientes, y los términos k-partito y k-coloreable tienen el mismo significado.

1.2. Polinomio Cromático

El **polinomio cromático** cuenta el número de maneras en las cuales puede ser coloreado un grafo usando no más que un número de colores dado.Por ejemplo, usando 3 colores, el grafo en la figura 2 puede ser coloreado de 12 formas distintas.Con solo 2 colores, no puede ser coloreado.Con 4 colores, puede ser coloreado de 24 + 4 x 12 maneras distintas: usando los cuatro colores juntos, hay

$4! = 24$ coloraciones válidas (toda asignación de cuatro colores a algún grafo de cuatro vértices es una coloración propia); y para cada elección de tres de los cuatro colores, hay 12 3-coloraciones válidas.

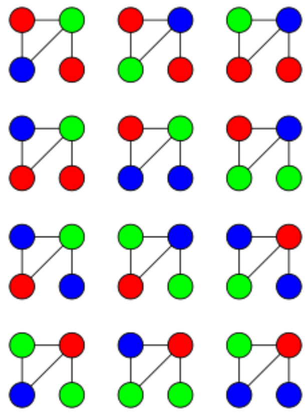


Figura 2: Coloración de un grafo de 4 vértices

El polinomio cromático es una función $P(G,t)$ que cuenta el número de t-coloraciones de G. como el nombre lo indica para un grafo G la función es un polinomio en t. para el grafo del ejemplo, $P(G,t) = t(t-1)^2(t-2)$ y $P(G,4) = 72$.

1.3. Arista Coloración

Una **arista coloración** de un grafo, es una coloración de las aristas, denotada como la asignación de colores a aristas tal que aristas incidentes tengan un color distinto. Una arista coloración con k colores es llamada k-arista-coloración y es equivalente al problema de particionar el conjunto de aristas en k emparejamientos. El menor número de colores necesarios para un arista coloración de un grafo G es el índice cromático o número cromático de aristas.

2. Sudoku como un problema de coloración

Sudoku está estrechamente relacionado con la teoría de grafos, ya que este rompecabezas puede replantearse como un problema de coloración de grafos, que consiste en la asignación de colores a los vértices de un grafo de manera que no haya dos vértices adyacentes que tengan el mismo color.

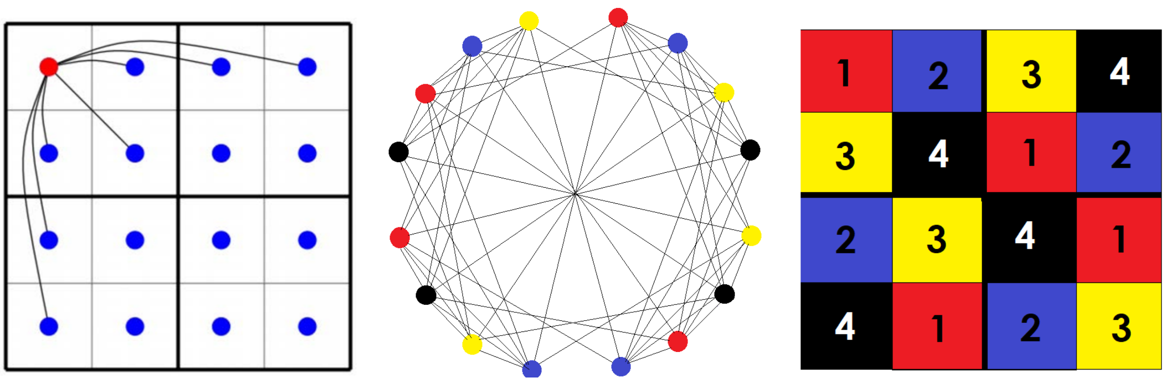
2.1. Mapeando un sudoku a un grafo

Un sudoku de rango n es un rompecabezas con n filas y n columnas de nxn celdas. El sudoku más común es de rango 3.
Para nuestro análisis, podemos definir una **zona** como un subconjunto de las n^4 celdas de un tablero de sudoku de modo que todos los miembros de la misma zona estén en la misma fila, columna o caja. Así el tablero de 9x9 tiene 27 zonas.
Cualquier rompecabeza de sudoku puede ser mapeado a un único grafo Sud(n)= (V,E) donde cada vértice de Sud(n) está conectado a todos los vértices con los que comparte una zona. De este modo, cada vértice esta conectado a n^2-1 otras celdas en su fila, columna y caja, de un total de $3(n^2-1)$. Pero $n-1$ celdas se cuentan como celdas adyacentes de fila y caja, y otros $n-1$ se cuentan como celdas adyacentes de columna y caja, por lo que el grado de cada vértice en el grafo sera $3(n^2-1)-2(n-1) = 3n^2-2n-1$. Luego, el grafo obtenido contiene n^4 vértices y $(3n^2-2n-1)(n^4)/2$ aristas. Decimos que Sud(n) es un grafo regular, ya que todos los vértices tienen el mismo grado.
Si, de la cuadricula, etiquetamos una celda ubicado en la i-esima fila y en la j-esima columna como (i,j), entonces dos vertices, que representan a dos celdas en el grafo Sud(n) etiquetados como (i,j) y (i',j'), estarán conectados si cumplen una de las siguientes condiciones: $i = i'$, $j = j'$, o $\lceil i/n \rceil = \lceil i'/n \rceil$ y $\lceil j/n \rceil = \lceil j'/n \rceil$.

j/i	1	2	3	4	5	6	7	8	9
1			1						
2			2		3				4
3				5			6		7
4	5			1	4				
5		7						2	
6					7	8			9
7	8		7			9			
8	4				6		3		
9							5		

Figura 3

Ahora el problema de resolver un sudoku es equivalente a encontrar una coloración adecuada del grafo Sud(3) con una paleta de 9 colores, dado la coloración previa de un subconjunto de los vértices del gráfo, ya que dos celdas tienen una arista entre sus vertices correspondientes solo si no son del mismo "color" o, en nuestro caso, número.



(a) (b) (c)

Figura 4: Proceso de solución de un sudoku de 4x4 celdas

De la figura 4, en (a) se muestra las conexiones de un vértice con los otros vértices que pertenecen a su "zona", por las reglas del juego los demás vértices que pertenecen a su fila, columna y caja deberán ser de diferente color a este vértice. Al realizar repetidas veces estas conexiones para cada vértice representado, obtenemos el grafo que se muestra en (b), llegado aquí se debe encontrar una coloración adecuada que satisfaga las observaciones ya mencionadas. En (c) se muestra el regreso del grafo a las cuadriculas, obteniéndose así la solución al sudoku.
Con motivo de responder a la pregunta de como sería la estructura de un grafo de un sudoku de rango 3, se muestra la siguiente figura.

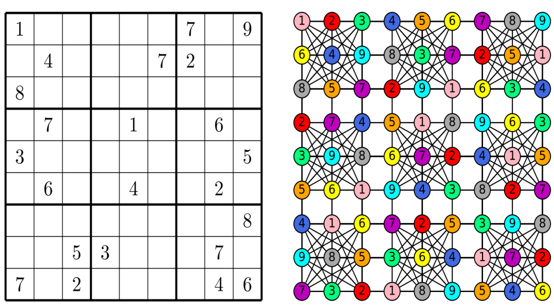


Figura 5: Sudoku resuelto por el grafo Sud(9) coloreado

La solución al problema reformulado del sudoku se plantea mediante un algoritmo presentado en la siguiente sección.

2.2. Algoritmo planteado

Para resolver el sudoku se realizan los siguientes pasos:

1. Leer el sudoku y generar el grafo con sus aristas y vértices coloreados de las casillas que ya tengan números.

2. Leer el grafo y colorear las aristas del mismo color que del vértice (no es necesario recolorear las aristas).

3. Leer el grafo y buscar vértices donde coincidan 8 aristas de diferente color y colorearla del color que falta(Continuar con el paso 4 Si se terminó de colorear los 81 vértices, sino volver al paso 2).

4. Rellenar el sudoku con el grafo coloreado obtenido.

2.3. Código de la aplicación en Python

Para la solución del sudoku se uso el siguiente código. Previamente descargar los archivos sudoku_gen.py y sudoku_color.py de (*Referencias 1*) necesarios para la generación y coloración del sudoku respectivamente.

```
from sudoku_color import *
from sys import setrecursionlimit

print("")
print("Continuemos con la solucion")
print("")
print("Pegar aqui!!   (enter para continuar)")
print("")
setrecursionlimit(100)
read_puzzle()
print("")
print("Resolviendo....")
print("")
for soln in color_puzzle(None, False):
    print_solution(soln)
    print("")
```

3. Aplicación de la teoría de grafos en horarios universitarios

Una de las aplicaciones de la coloración de grafos es la de ordenar cursos de algún centro educativo, los cuales están sujetos a restricciones, intervalos, etc.
En la asignación de horarios se pueden reconocer diversas variables.

1. Las clases

2. Número de aulas

3. Intervalos de cada clase

4. Restricciones:

    a. Unitarias:
    No hay quien dicte la clase de redacción los días martes.
    La clase de álgebra lineal debe durar 2 horas.

b. Binarias:
Cualquier par de clases no pueden dictarse a la misma hora.

c. De capacidad:
Las aulas R son muy pequeñas para la clase de cálculo.

d. De agente:
El profesor debe tener 3 mañanas libres por semana.

Las restricciones se dividen en fuertes y débiles. Por ejemplo, una restricción tipo unitaria tiene más prioridad que una de tipo agente.
La restricción binaria evento conflictivo es una de las más importantes. Existen varios casos de este conflicto. Uno de ellos indica que si un alumno debe estar en dos clases, estas no pueden ocurrir al mismo tiempo.
Esta restricción permite establecer una conexión entre la asignación de horarios y la coloración de grafos. Las clases serán los vértices, los eventos conflictivos serán las aristas y los intervalos de tiempo, colores.

3.1. Descripción del problema

Crear un horario para cierta cantidad de clases, en donde se eviten a toda costa las restricciones fuertes. Este problema involucra cuatro tipos de restricciones fuertes; es decir, su cumplimiento es obligatorio.

3.2. Características/restricciones del problema

1. Las clases deben ser asignadas en intervalos de forma que ningún alumno tenga que estar en más de una al mismo tiempo.
2. Asistencia: Para cada clase, hay un número de estudiantes que se han matriculado y tienen la intención de asistir.
3. Propiedades de aulas: Cada aula tiene un número determinado de carpetas, computadoras, etc.
4. Características de clase: Cada clase deberá ser asignada en un aula que cumpla con las condiciones necesarias.
5. Restricción de disponibilidad: Algunas clases no pueden ser dictadas en ciertos intervalos.

3.3. Solución del problema

Sea el conjunto de clases:

$$E = \{e_1, e_2, ..., e_n\}$$

Sea el conjunto de intervalos:

$$T = \{t_1, t_2, ..., t_{45}\}$$

Sea el conjunto de alumnos:

$$S = \{s_1, s_2, ..., s_s\}$$

Sea el conjunto de aulas:

$$R = \{r_1, r_2, ..., r_r\}$$

cada aula tiene asociado un $C(r_i)$ que representa el número de carpetas u ordenadores de cada aula.

Y sea el conjunto de características asociadas:

$$F = \{f_1, f_2, ..., f_f\}$$

Ahora a cada conjunto le asociamos una matriz:

Matriz de asistencia: $\mathbb{R}^s \to \mathbb{R}^n$

$$P_{ij}^{(1)} = \begin{cases} 1 & \text{el estudiante } s_i \text{ debe asistir a la clase } e_j \\ 0 & \text{caso contrario} \end{cases}$$

Matriz de propiedades de aula: $\mathbb{R}^r \to \mathbb{R}^f$

$$P_{ij}^{(2)} = \begin{cases} 1 & \text{el aula } r_i \text{ tiene la característica } f_j \\ 0 & \text{caso contrario} \end{cases}$$

Matriz de características de clase: $\mathbb{R}^n \to \mathbb{R}^f$

$$P_{ij}^{(3)} = \begin{cases} 1 & \text{la clase } e_i \text{ requiere la característica } f_j \\ 0 & \text{caso contrario} \end{cases}$$

Matriz de disponibilidad de clases: $\mathbb{R}^n \to \mathbb{R}^k$

$$P_{ij}^{(4)} = \begin{cases} 1 & \text{la clase } e_i \text{ puede ser asignada al intervalo } t_j \\ 0 & \text{caso contrario} \end{cases}$$

Además de las cuatro matrices ya nombradas, añadiremos dos matrices más, de forma que podamos detectar más rápidamente el incumplimiento de restricciones fuertes.

Matriz de idoneidad del aula: $\mathbb{R}^n \to \mathbb{R}^r$

$$R_{ij} = \begin{cases} 1 & \left(\sum_{l=1}^{s} P_{li}^{(1)} \leq c(r_j)\right) \wedge \left(\nexists f_l \in f/(P_{il}^{(3)} = 1 \wedge P_{jl}^{(2)} = 0)\right) \\ 0 & \text{caso contrario} \end{cases}$$

La matriz R especifica las aulas que cumplen con las condiciones necesarias para cada evento.

$$\left(\sum_{l=1}^{s} P_{li}^{(1)} \leq c(r_j)\right)$$

En esta expresión se hace lo siguiente:

Estamos sumando todos los elementos de cada columna de la matriz $P^{(1)}$. Recordemos que en esta matriz se ve la relación entre alumnos y la clase a la que deben asistir; es decir, la suma de los elementos de una columna de esta matriz representa el número de alumnos que habrá en cada clase. Si este número es menor o igual a la capacidad del aula (se están evaluando en todas las aulas), quiere decir que esa clase puede dictarse en esa aula

$$\left(\nexists f_l \in f/(P_{il}^{(3)} = 1 \wedge P_{jl}^{(2)} = 0)\right)$$

En esta expresión se hace lo siguiente:

Estamos descartando las características que cumplan que una clase requiera de dicha característica (eso es lo que hace la matriz $P^{(3)}$), pero que un aula no tenga dicha característica ($P^{(2)}$).

Matriz de conflictos: $\mathbb{R}^n \to \mathbb{R}^n$

$$C_{ij} = \begin{cases} 1 & \left(\exists s_l \in s / \left(P^{(1)}_{li} = 1 \wedge P^{(1)}_{lj} = 1\right)\right) \\ & \vee \left((\exists r_l \in r / (R_{il} = 1 \wedge R_{jl} = 1)) \wedge \left(\sum_{l=1}^{r} R_{il} = 1\right) \wedge \left(\sum_{l=1}^{r} R_{jl} = 1\right)\right) \\ & \vee \left(\nexists t_l \in t / \left(P^{(4)}_{il} = 1 \wedge P^{(4)}_{jl} = 1\right)\right) \\ 0 & \text{caso contrario} \end{cases}$$

La matriz C es una matriz simétrica que especifica pares de eventos que no pueden ser asignados al mismo intervalo (eventos conflictivos).

Esto puede ocurrir si: Dos clases e_i y e_j

1. Tienen un estudiante en común.
2. Necesitan impartirse en la misma aula.
3. No pueden suceder simultáneamente con respecto a los intervalos para los que están disponibles.

Esta matriz C es la matriz de adyacencia de un grafo $G = (V, E)$ con n vértices. Sin embargo, a diferencia del problema de coloración de grafos, en este caso, el orden de los intervalos, que se corresponde con las clases de color, influye en gran medida a la solución(no es lo mismo llevar primero cálculo y luego lineal que viceversa). Es por eso que una solución al problema de horarios se representa como un conjunto ordenado de conjuntos $S = \{S_1, ..., S_{k=45}\}$ sujeto a las siguientes restricciones fuertes:

1. S debe particionar el conjunto de clases E (o un subconjunto de este), en un conjunto ordenado de conjuntos, etiquetados como $S_1, ..., S_k$. Cada conjunto $S_i \in S$ contiene eventos asignados al intervalo t_i.

$$\bigcup_{i=1}^{k} S_i \subset E$$

$$S_i \cap S_j = \emptyset \ (1 \leq i \neq j \leq k)$$

2. Ningún par de clases conflictivas deben ser asignadas al mismo conjunto $S_i \in S$.

$$\forall e_j, e_l \in S_i, C_{jl} = 0 \ (1 \leq i \leq k)$$

3. Cada clase debe ser asignada a un conjunto $S_i \in S$ cuyo intervalo correspondiente t_i está disponible en base a la matriz $P^{(4)}$.

$$\forall e_j \in S_i, P^{(4)}_{ji} = 1 \ (1 \leq i \leq k)$$

4. Garantizar que las clases asignadas a un conjunto $S_i \in S$ pueden ser asignadas a un aula apropiada del conjunto de aulas R. Para que esto sea posible, es necesario resolver un problema de maximización de correspondencias bipartitas.
Sea $G = (S_i, r, E)$ un grafo bipartito con S_i y r como conjuntos de vértices y un conjunto de aristas $E = \{\{e_j \in S_i, r_l \in r\} \ / \ R_{ij} = 1\}$. Dado el grafo G, el conjunto S_i es un elemento de $\mathcal{M}$ si y sólo si existe una correspondencia bipartita máxima de G de tamaño $2|S_i|$, donde en cada caso, las restricciones asociadas a cada aula para ese intervalo deben cumplirse.

$$S_i \in \mathcal{M} \ (1 \leq i \leq k)$$

Una solución S se considera válida si y sólo si se verifican todas estas restricciones. La calidad de una solución valida se mide en base a la distancia a factibilidad (DTF), medida que se calcula como la suma de los tamaños de todos los eventos que no están presentes en la solución.

$$DFT = \sum_{e_i \in S'} \sum_{j=1}^{|s|} P_{ij}^{(1)}$$

donde $S' = E - \bigcup_{i=1}^{k} S_i$. Si la solución S es válida y tiene DTF cero; es decir, $E = \bigcup_{i=1}^{k} S_i$ y equivalentemente $S' = \emptyset$, entonces S se considera factible, pues todas las clases han sido asignadas en un intervalo de forma factible.
Al tener nuestra matriz de conflictos ya hecha, nos percatamos de algo. Los vértices del grafo al que esta matriz está asociada vendrían a ser las clases, y estas clases están unidas por aristas si, como dice el nombre de la matriz, hay un conflicto entre ellas. Esto quiere decir que no pueden ir en el mismo intervalo. Aquí es donde entra la coloración de grafos, pues, como sabemos, la idea de la coloración es que dos vértices que se encuentran unidos deben tener diferente color, que para nuestro problema de horarios, significa intervalo.

"Diferentes intervalos para las clases que presenten conflictos, diferentes colores para los vértices unidos por una arista."

3.4. Ejemplo:

Se tiene el siguiente conjunto de clases, donde T indica que es la parte teórica y P la parte práctica:

1. Programación orientada a objetos (T)
2. Programación orientada a objetos (P)
3. Cálculo Avanzado (T)
4. Cálculo Avanzado (P)
5. Análisis Real II (T)
6. Análisis Real II (P)
7. Física II (T)
8. Física II (P)
9. Inglés (T)

$$E = \{e_1, e_2, ..., e_9\}$$

Se tiene el siguiente conjunto de intervalos:

1. 08:00 - 10:00
2. 10:00 - 12:00
3. 14:00 - 16:00
4. 16:00 - 18:00

Por día, obviamente.

$$T = \{t_1, t_2, t_3, t_4, ..., t_{20}\}$$

Se tiene el siguiente conjunto de alumnos:

1. 15 alumnos de la carrera de Ciencias de la Computación
2. 20 alumnos de la carrera de Matemática
3. 15 alumnos de la carrera de Física

$$S = \{s_1, s_2, ..., s_{50}\}$$

Se tiene el siguiente conjunto de aulas:

1. Dos aulas R con capacidad para 20 alumnos
2. Un aula J con capacidad para 50 alumnos
3. Una sala de computadoras con capacidad para 15 alumnos

$$R = \{r_1, r_2, r_3, r_4\}$$

$$C(r_1) = 20; C(r_2) = 20; C(r_3) = 50; C(r_4) = 15$$

Se tiene el siguiente conjunto de características:

1. Tiene computadoras

$$F = \{f_1\}$$

Restricciones:

1. De asistencia:
   - *a.* Los alumnos de CC deben estar matriculados en los cursos de POO, Cálculo A. e Inglés.
   - *b.* Los alumnos de Matemática deben estar matriculados en los cursos de Cálculo A. y Análisis R. II
   - *c.* Los alumnos de Física deben estar matriculados en Cálculo A., Física II.
2. De propiedades de aula:
   - *a.* Solo la sala tiene computadoras.
3. De características de clase:
   - *a.* La clase de POO necesita computadoras.
4. De disponibilidad de clases:
   - *b.* La clase de práctica de Física II debe ser dictada en la mañana.
   - *c.* La clase teórica de Análisis Real II debe ser dictada en la tarde.

Comencemos con las matrices:

$$P^{(1)} = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

Las 15 primeras filas son los alumnos de CC, las 20 siguientes son los de Matemática, y las 15 últimas son los de Física.

$$P^{(2)} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$P^{(3)} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$P^{(4)} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

Con estas matrices hallamos nuestra matriz R:

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

Y ahora la matriz C:

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

En este caso, debemos poner sí o sí ceros en la diagonal, pues con las condiciones a evaluar si salen 1, pero entonces sería un grafo con bucles.
Esta matriz C es la matriz de adyacencia del siguiente grafo:

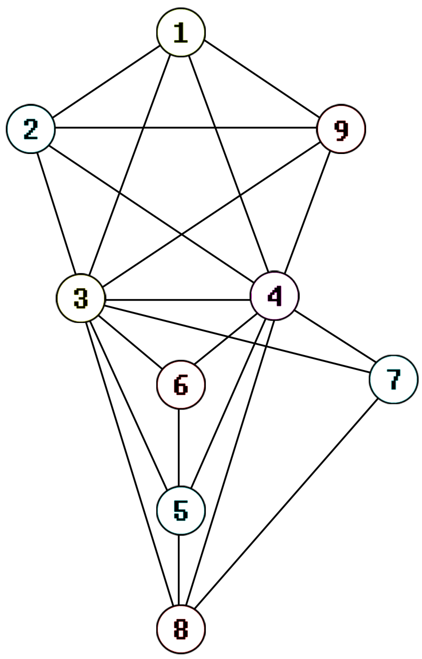


Figura 6

Y aquí presentamos una de las muchas colorizaciones para dicho grafo:

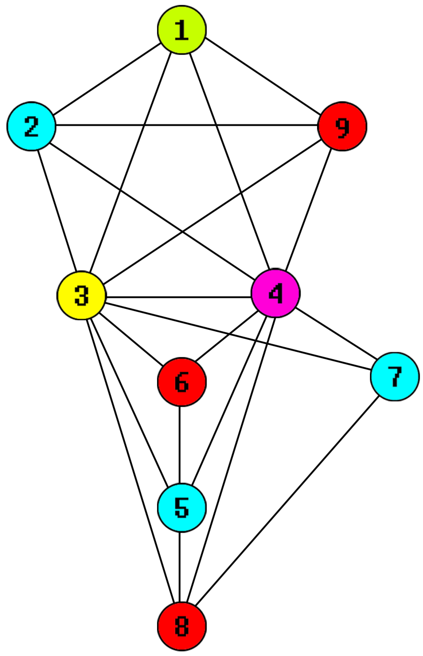


Figura 7

¿Como elegimos que color le corresponde a que intervalo?

Lo elegimos respetando las restricciones que pusimos anteriormente

1. Física II (P) debe ser dictado en la mañana, el cual es el número 8, entonces el color correspondiente a ese número debe ser un intervalo de la mañana. Así que podemos escoger entre los intervalos 1 o 2.

2. Análisis Real II (T) debe ser dictado en la tarde, el cual es el número 5, entonces el color correspondiente a ese número debe ser un intervalo de la tarde. Así que podemos escoger entre los intervalos 3 o 4.

Uno de los horarios que genera esta coloración es el siguiente:

Horario de universidad				
	Lunes			Martes
08:00 - 10:00	Física II (P)	Análisis R. II (P)	Inglés (T)	
10:00 - 12:00	POO (T)			
14:00 - 16:00	Análisis R. II (T)	POO (P)	Física II (T)	Cálculo A. (T)
16:00 - 18:00	Cálculo A. (P)			

Como podemos apreciar, el horario cumple con las restricciones que nombramos.

3.5. Código de la aplicación en Python

Para la generación del horario se uso el siguiente código. Previamente descargar los archivos Horario_util.py que contiene las funciones usadas y Horario_color.py necesario para la coloración del grafo, del enlace en (*Referencias 1*).

```
from Horario_util import *
from Horario_color import *
from os import system

#Obtenemos la informacion del usuario
E = getEvents()
T = getTimeIntervals()
S = getStudents()
R = getRooms()
```

```
F = getCaracteristics()
#Preguntamos por las restricciones y generamos P1 P2 P3 P4
P1 = genP1(S,E)
P2 = genP2(R,F)
P3 = genP3(E,F)
P4 = genP4(E,T)
system('cls')     #limpiando la pantalla
#generamos r y c
r = genR(P1,P2,P3)
c = genC(P1,P4,r)          #finalmente c sera la matriz de adyacencia
print("")      #salto de linea
print("Matriz de Adyacencia")
print("")      #salto de linea
Print(c,len(E),len(E))     #La imprimimos para visualizarla
#Procedemos con la coloracion de los vertices del grafo
Vertex = colorG(c)         #Obtenemos los vertices coloreados
print("")      #salto de linea
print("Vertices coloreados")
print("")      #salto de linea
for i in range(len(Vertex)):
    print(Vertex[i], end=" ")
print("")      #salto de linea
print("Con los vertices ya coloreados procedemos a generar el horario")
print("")      #salto de linea
l = input("Enter para continuar")
system('cls')
#Procedemos a imprimir el Horario
PrintHorario(Vertex,P4)
```

4. Referencias

1. https://github.com/wencez432/Coloraci-n-de-Grafos
2. https://www.scielo.sa.cr/pdf/rmta/v18n2/a02v18n2.pdf
3. http://eio.usc.es/pub/mte/descargas/ProyectosFinMaster/Proyecto_1463.pdf
4. https://programacion.net/articulo/problema_de_asignacion_de_horarios_271
5. http://web.mit.edu/course/other/sp.268/OldFiles/www/boardgames.pdf
6. https://hlma.math.cuhk.edu.hk/wp-content/uploads/2018/07/e76c6ddf6ceafb687d3f500ff691d593.pdf
7. https://www.youtube.com/watch?v=em5Y-Z_TFyo&t=314s
8. A Guide to Graph Colouring.Algorithms and Applications, Lewis, R.M.R., 2015