Universidad de Santiago de Chile

Facultad de Ingeniería

Departamento de Ingeniería Informática

Tarea N°1: Algoritmos de coloración de Grafos

Integrante: Matías Cortés.

Ignacio Villarroel.

Profesor: Cristián Sepúlveda.

Ayudante: Cesar Rivera.

Introducción

La Teoría de Grafos tienes sus primeras ideas en el siglo XVIII, dichas por Leonhard Euler, en la cual planteaba el *Problema de los siete puentes de Königsberg*, el problema consistía en encontrar un camino tal que, pasando una única vez por cada uno, se pudiera regresar a al punto de partida. La respuesta a esto era de que no existe un camino que cumla con estas características, su demostración se basó en el que los puentes eran aristas, en la cual estos conectaban los vértices que representaban a las regiones de la ciudad. Esto marcó el inicio de la Teoría de Grafos.

Imagen 1. Representación problema de los 7 puentes de Königsberg.

Posterior a esto en el año 1852 Francis Guthrie, explicó sobre la coloración de mapas de Inglaterra, donde se dio cuenta que los condados que tenían contacto entre ellos (vecinos), era solo necesario usar 4 colores para asegurar que todos los condados vecinos se le asignaban colores distintos, esto dio inicio al “Problemas de coloración de Grafos”.

Descripción del problema

El problema seleccionado es sobre la coloración de grafos. Esta problemática se basa en comprobar todas las posibles asignaciones entre colores y vértices para escoger la mejor de todas ellas, siendo dado la cantidad de colores a utilizar. La real complejidad de esto es cuando aumenta considerablemente la cantidad de vértices (“n” vértices), esto hace que de la misma forma aumente considerablemente la cantidad de colores (“n” colores), de esta forma se tendrá una cantidad de n^n posibles soluciones esto genera que llega hasta un punto donde ni los mejores computadores pueden resolverlo, sin embargo, existen excepciones donde se puede encontrar la solución óptima a esta clase de problemas, para ellos existen algoritmos que recorren los posibles caminos, eso sí, siempre considerando la cantidad de vértices y colores.

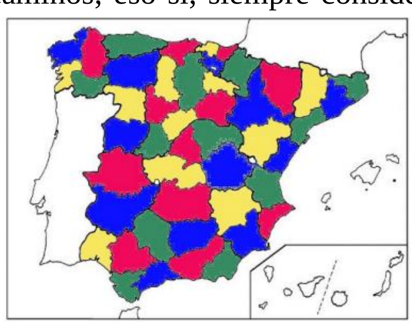

Imagen 2. Coloración de provincias de España.

Descripción de Algoritmos

Se eligieron 2 algoritmos que cumplen con las características de que pueden resolver el problema, se procederá a explicar cada uno.

a) Algoritmo Voraz:

Consta de los algoritmos más simples y a su vez es uno de los algoritmos heurísticos mas importantes en lo que respecta a coloración de grafos, también conocido como “Greedy”. La estrategia principal de este algoritmo se basa en ir tomando vértices del grafo uno por uno, siguiendo un orden (aleatorio) y asignando a cada vértice el primer color disponible. Así sucesivamente pasando por todos los vértices, sin embargo, la solución no necesariamente tiene que ser la óptima.

Solamente una correcta elección en el orden de los vértices para su coloración puede producir una solución óptima para cualquier grafo. En la práctica del algoritmo se indica que genera soluciones factibles de forma rápida, aunque como se mencionó no son las mejor de todas las opciones. A nivel de pseudocódigo tenemos que:

```
GREEDY (S ← ∅, π)
for i ← 1 to |π| do
    for j ← 1 to |S|
        if (S_j ∪ {π_i}) is an independent set then
            S_j ← S_j ∪ {π_i}
            break
        else j ← j + 1
    if j > |S| then
        S_j ← {π_i}
        S ← S ∪ S_j
```

Respecto a las propiedades del algoritmo en el peor de los casos para establecer la coloración de un grafo la complejidad es de orden $O(n^2)$ siendo para un peor caso bastante competente aunque no es lo mejor.

b) Algoritmo RLF (Recursive Largest First):

Es mas complejo que el anterior, su funcionamiento se basa en ir coloreando el grafo con color por cada iteración del algoritmo, en vez de un vértice por iteración, es decir en cada iteración el algoritmo busca conjuntos de vértices independientes en el grafo los cuales se les asociará un mismo color, ya que no tendrán interacción entre si, luego este conjunto será eliminado del grafo y se procederá de la misma forma con un subgrafo restante, hasta que el subgrafo creado sea vacío lo que indicará que todos los vértices tienen asociado un color. A nivel de pseudocódigo se muestra lo descrito:

```
RLF (𝒮 ← ∅, X ← V, Y ← ∅, i ← 0)
while X ≠ ∅ do
    i ← i + 1
    S_i ← ∅
    while X ≠ ∅ do
        Choose v ∈ X
        S_i ← S_i ∪ {v}
        Y ← Y ∪ Γ_X(v)
        X ← X − (Y ∪ {v})
    𝒮 ← 𝒮 ∪ {S_i}
    X ← Y
    Y ← ∅
```

Sobre las propiedades se indica que la complejidad corresponde a $O(n^3)$ lo que hace dar cuenta que esta clase de algoritmo genera un mayor esfuerzo a nivel computacional en lo que respecto a algoritmo voraz mostrado anteriormente.

Descripción de Aplicación

La utilidad que provee la solución a esta clase de problema tiene un campo de aplicación, por lo que se eligió una en particular

a) Sudoku:

El clásico juego donde no se puede repetir un número en una fila, columna y en un conjunto cuadrado, es efectivamente utilizable, explicando de manera algorítmica donde cada cuadro se representa como un vértice, donde cada fila y columna contenga los caracteres 1, 2, 3… n, una única vez, además la s rejillas están divididas en “n” cajas las cuales nuevamente deben contener caracteres del 1, 2, 3… n, también una única vez. De esta forma podemos crear un problema de coloración donde cada carácter representará el color y se de cumplir específicamente las reglas entregadas.

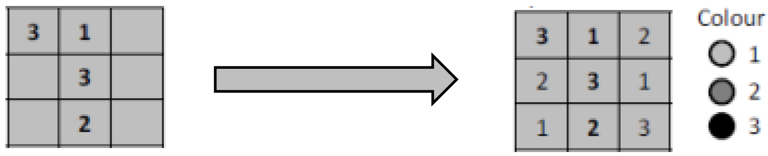

Imagen 3. Resolución Sudoku por Coloración

<u>Referencias</u>

- Sergio Pena Seijas. (2016, marzo). *El Problema de Coloración de Grafos*. Universidad de Vigo.
- Murga Díaz, M. (2013, febrero). *https://repositorio.unican.es/xmlui/bitstream/handle/10902/3099/Maria%20Rosa%20Murga%20Diaz.pdf?sequence=4*. Universidad de Cantabria. https://repositorio.unican.es/xmlui/bitstream/handle/10902/3099/Maria%20Rosa%20Murga%20Diaz.pdf?sequence=4