Coloreo de Grafos con Simulated Annealing

Ramiro Hoffens
Benjamin Mitchell

Coloreo de Grafos

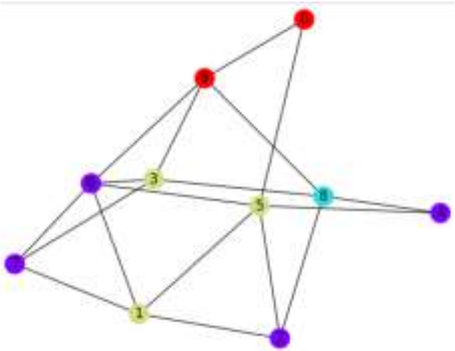
- Sean $G = (V, E)$ un grafo y C un conjunto de colores.
- Se plantea el problema de coloreo de grafos, que consiste en asignarle colores de C a los nodos del grafo de forma que no hayan nodos adyacentes con el mismo color.
- La idea es resolverlo para C lo más pequeño posible.
- Se define el número cromático $(\chi(G))$ como la cantidad mínima de colores con los que es factible resolver el problema. Consideramos la cota $\chi(G) \leq \Delta(G)+1$.
- Es un problema *NP-completo*.
- El algoritmo de coloreo de grafos de la librería *NetworkX* pide cantidad de colores mayor a la cota antes mencionada (lo cual podría no ser óptimo).

Coloreo de Grafos

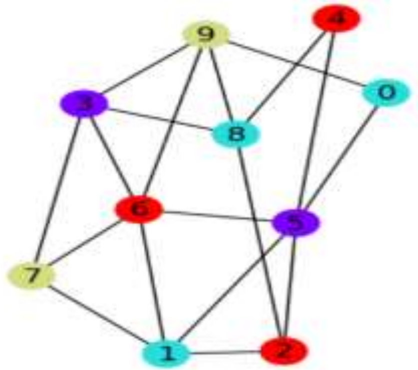
El problema de coloreo puede ser planteado como uno de minimización sobre la función:

$$H(x) = \sum_{v \sim u} \mathbf{1}_{x_v = x_u}$$

Donde un coloreo x es solución $\Leftrightarrow$ H(x)=0.



$\Rightarrow$

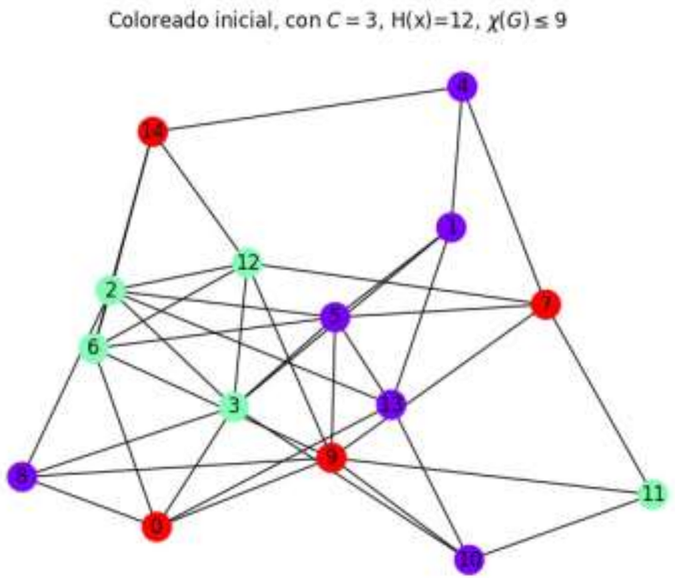


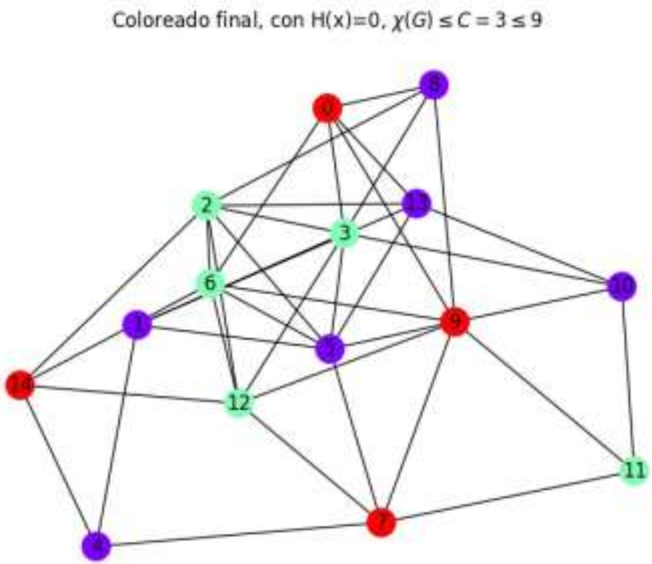
Simulated Annealing

- Al plantear el problema como uno de optimización sobre la función de costos *H(x)*, podemos usar un algoritmo estocástico para resolverlo. Para este caso, implementaremos el algoritmo de Metropolis-Hasting
- El algoritmo de metrópolis define una cadena de Markov sobre las configuraciones del coloreado, donde de un estado inicial se salta a otro tal que se priorizan los saltos que hacen decrecer *H*.
- Generamos un grafo aleatorio de *n* nodos y aristas con probabilidad *p*, asignamos colores a cada nodo de forma aleatoria
- Queremos buscar el buen coloreado, y la cantidad mínima de colores

Aplicación 1: Coloreado de grafos

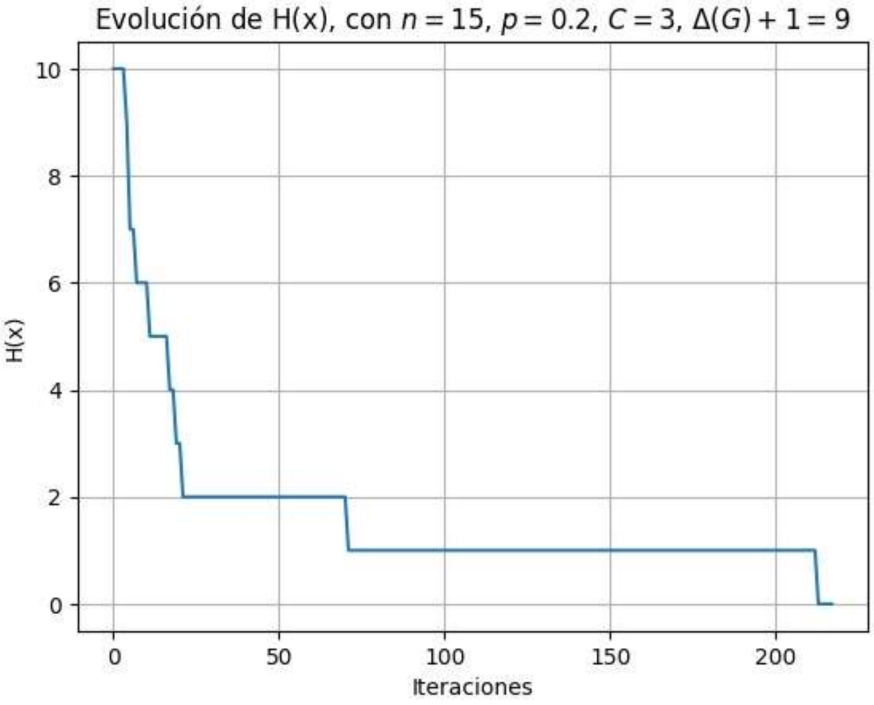
Dados G y C, se ejecuta el algoritmo Annealing para buscar un coloreado óptimo, la factibilidad no es segura cuando $|C| \leq \Delta(G)+1$, pero es de interés particular.

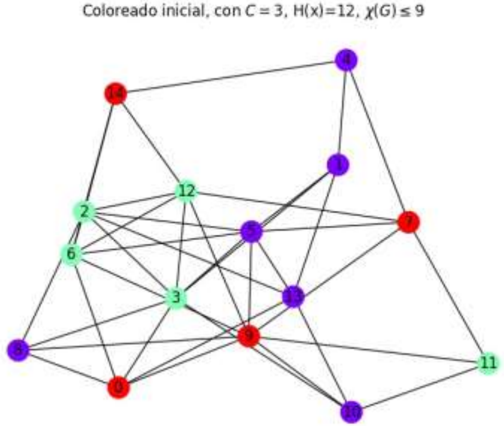


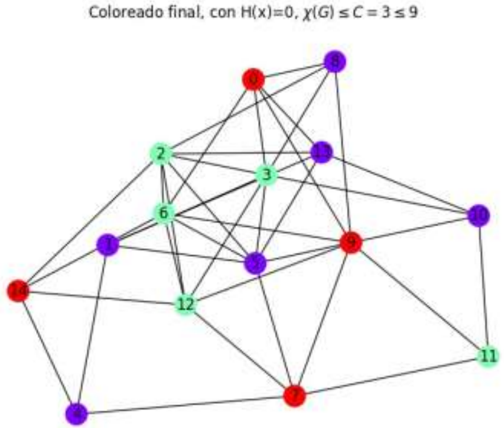


Aplicación 1: Coloreado de grafos

Convergencia del sistema *(k_sol~215)*:

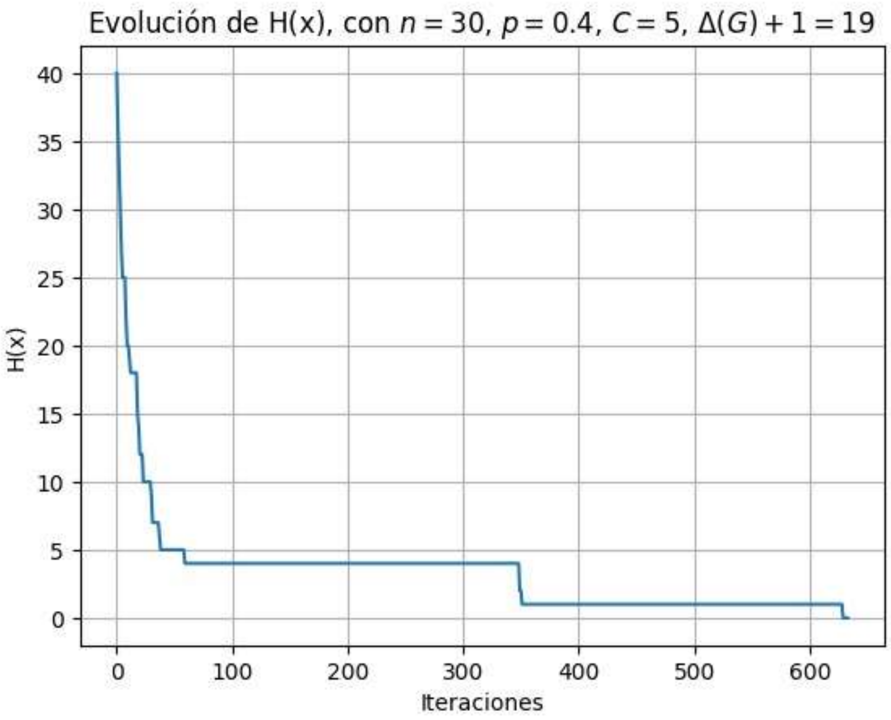


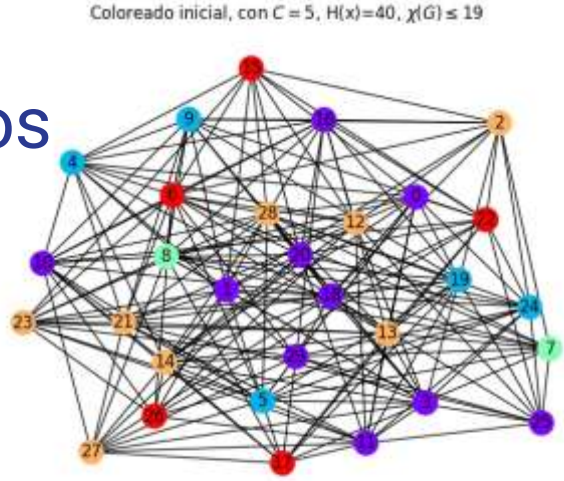


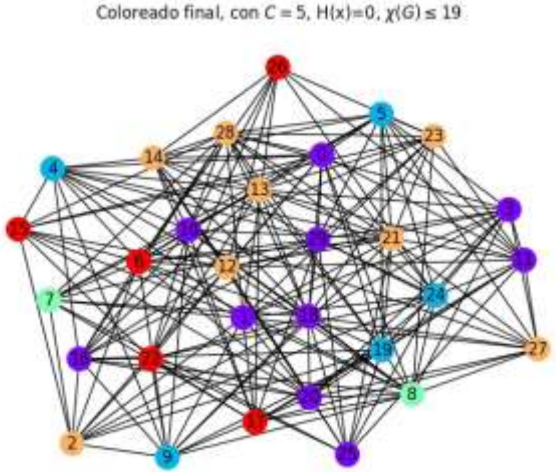


Aplicación 1: Coloreado de grafos

Segundo ejemplo ***(k_sol~650)***:





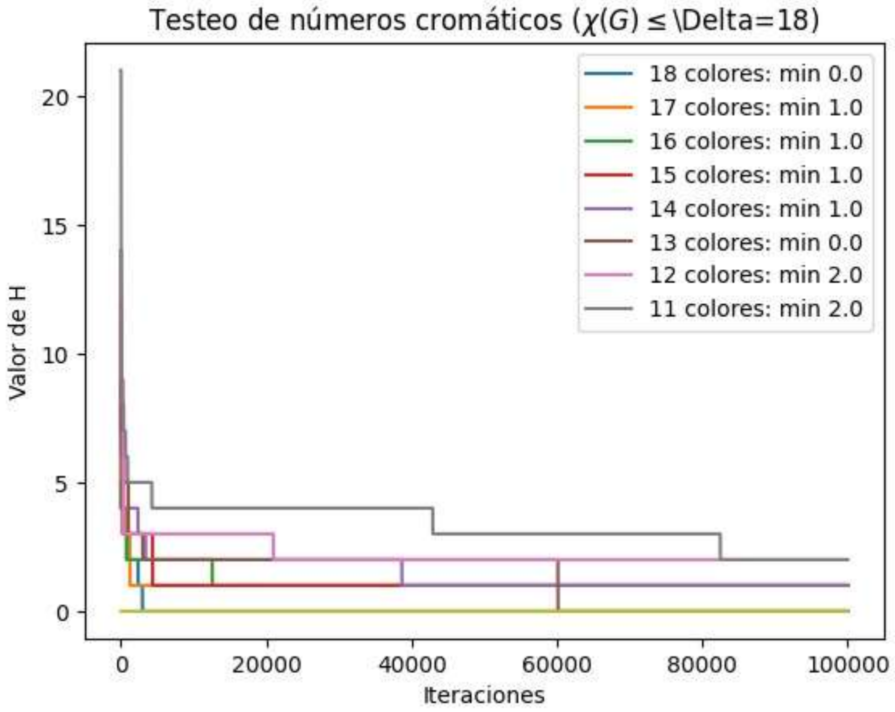


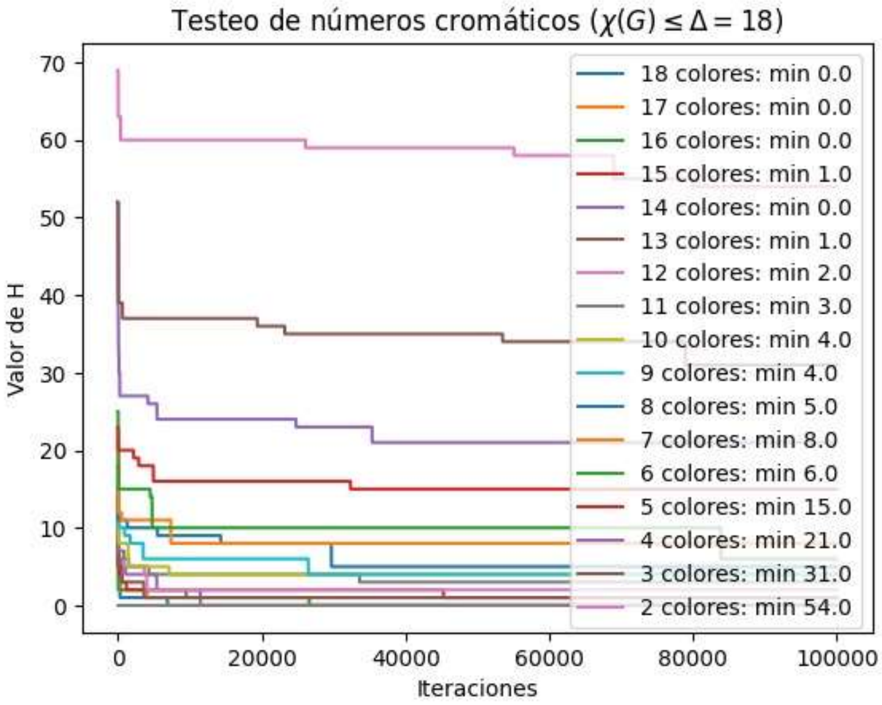
Aplicación 2: Refinar cotas para $\chi(G)$

En lo anterior se consiguieron coloreados óptimos con una cantidad de colores considerablemente menor a la cota teórica (3 y 9; 5 y 19). Esto permite mejorar la cota para el número cromático de un grafo dado (y agrandar el conjunto conocido factible).

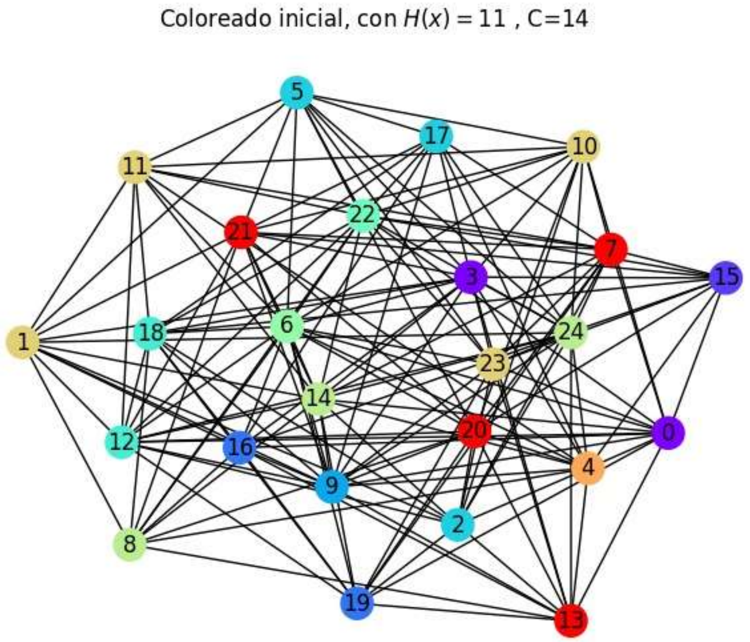
La segunda aplicación consiste en ejecutar Annealing distintas cantidades de colores, menores a la cota, con el objetivo de obtener cotas mejores para el número cromático.

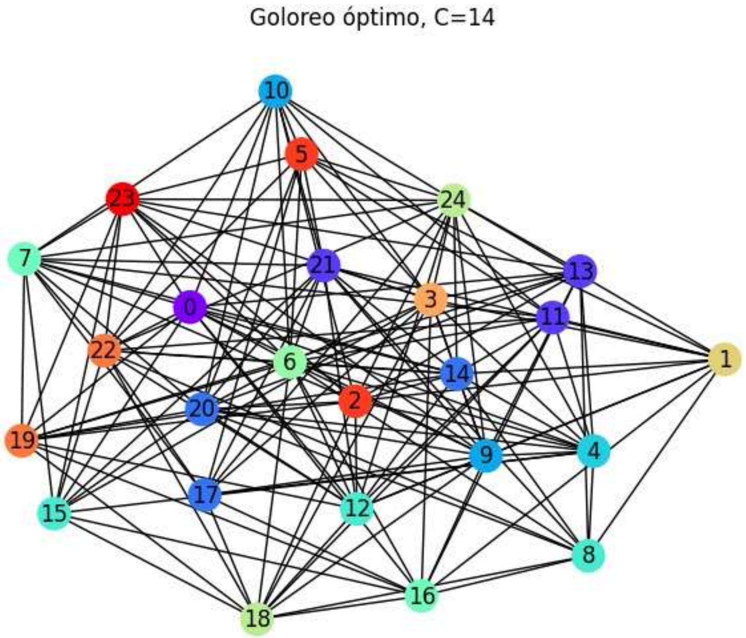
Aplicación 2: Refinar cotas para $\chi(G)$





Aplicación 2: Refinar cotas para $\chi(G)$





Conclusiones

Mediante la implementación de algoritmos aleatorios, fue posible resolver problemas de coloreo de grafos y estudiar el rango de colores factibles para su resolución. Y, si bien estos, al no ser deterministas, podrían detenerse en algún mínimo local o no converger a la solución en una cantidad reducida de pasos; al iterar e ir variando los parámetros es posible aproximar soluciones cada vez mejores para problemas que serían muy costosos de abordar de manera determinista.

Bibliografía

- *’Simulated Annealing Algorith for Graph Coloring’*; A. Köse, B. Aral, M. Balaban, 2017
- *’Apuntes Simulación Estocástica’*; Joaquín Fontbona, 2023
- *’Coloración de Grafos’*, María Rosa Murga Díaz, Universidad de Cantabria, 2013