

Simulated Annealing aplicado al coloreamiento de grafos

Rodrigo Altamirano M., Gonzalo Ovalle J.

20 de diciembre del 2022

Coloreamiento de un grafo

El coloreamiento en un grafo es un caso especial de etiquetado, en el cual dichas etiquetas son llamados colores a vértices del grafo. Una coloración *propia* de los vértices se da cuando los vértices adyacentes no comparten ningún color.

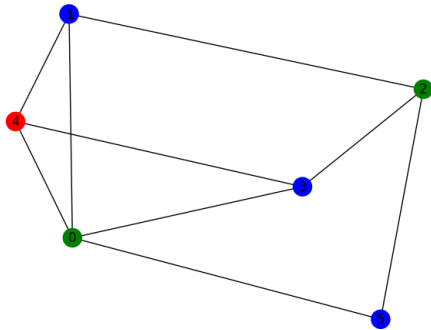


Figure 1: Ejemplo de un 3-coloreamiento de un grafo.

Problemática

El problema del coloreamiento de un grafo consiste en encontrar la coloración *propia* que utilice la menor cantidad de colores distintos. Esto es útil pues en diversos contextos industriales/teóricos la coloración de un grafo permite modelar relaciones y conexiones de un ambiente determinado, y la cantidad de colores podría representar algún recurso a optimizar. Se sabe que el problema de encontrar $X(G)$ en grafos generales es un problema NP-completo, por lo que intentaremos abordarlo mediante metaheurísticas de búsqueda local (LS), empleadas en 3 algoritmos distintos: El algoritmo Dsatur (un algoritmo *Greedy*), Simulated Annealing y con Tabu Search. El objetivo es comparar estos algoritmos según la calidad de sus soluciones (número de colores encontrados para el coloreamiento obtenido) en algunos tipos de grafos y también hacer un análisis respecto a la complejidad temporal práctica que toma ejecutar los algoritmos.

La implementación de Simulated Annealing para una cantidad de colores fija se hizo vía Metropolis y se utilizó la siguiente función de costo de un coloreamiento:

$$H(x) = \sum_{(v,w) \in E} 1_{x_v = x_w}$$

En donde se entrega 1 si dos vertices de una arista comparten color o 0 si no. Bajo esta lógica, minimizar esta función y que este mínimo sea 0 implica que el coloreamiento x es uno propio según la definición anterior.

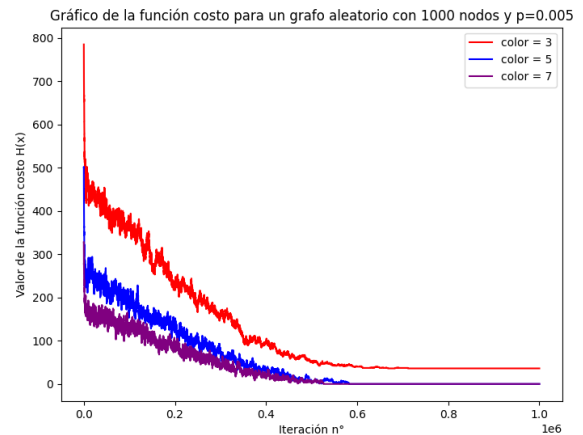


Figure 2: Gráfico de las funciones costo de Simulated Annealing para un grafo de Erdos-Renyi con grado promedio de vértice 5 y 1000 nodos. Se puede ver que para 3 colores $H_{min} = 36$, mientras que para 5 y 7 colores $H_{min} = 0$.

Bibliografía

1. Köse, A., Aral, B., Balaban, M. Simulated Annealing Algorithm for Graph Coloring.
2. Lewis, R.(2016).A guide to graph colouring.Springer.
3. Implementación de Tabu Search. https://github.com/MaciejPel/graph-coloring/blob/main/graph_coloring.py