# Proyecto final Simulación Estocástica

## Simulated Annealing aplicado al coloreamiento de grafos

Rodrigo Altamirano M., Gonzalo Ovalle J.

Universidad de Chile
Facultad de Ciencias Físicas y Matemáticas
Departamento de Ingeniería Matemática

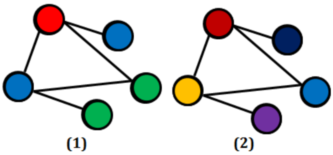
21 de diciembre de 2022

# Contenidos

# Problema

## Colorear un grafo

En este contexto, **colorear** un grafo no dirigido $G = (V, E)$ consiste una asignación de colores a sus vértices de manera que no hayan dos vértices adyacentes que compartan color. El problema del coloreamiento de un grafo consiste en encontrar la menor cantidad de colores con la cual poder colorear $G$. Dicha cantidad se llama Número cromático de $G$ denotada $X(G)$.
Se sabe que este problema es difícil de resolver en grafos generales, así que se buscaran coloreamientos vía metaheurísticas de busqueda local. (algoritmos).



# Algoritmo Greedy

## Grado de saturación

Sea $c(v) = NULL$ para todo vértice $v \in V$ que no tiene una clase de color asignada. El grado de saturación de un vértice v dado se define como el número de diferentes colores asignados a vétices adyacentes a v, es decir, $sat(v) = |\{c(u) : u \in \Gamma(v) \wedge c(u) \neq NULL\}|$.

## Conjunto independiente

Un conjunto independiente es un conjunto de vértices $I \subseteq V$ tal que sus elementos son mutuamente no adyacentes, esto es, $\forall u, v \in I, (u, v) \notin E$.

# Algoritmo Greedy

## DSATUR

DSATUR es un algoritmo del tipo glotón que se basa en tomar cualquier vértice y le asigna la primera clase de color válida o crea otra si es necesaria. Luego para escoger el siguiente vétice a colorear, se hace mediante una heurística que se basa en las características del coloreo actual del grafo. Esta elección se basa principalmente en el grado de saturación definido anteriormente.

```
DSatur (𝒮 ← ∅, X ← V)
while X ≠ ∅ do
    Choose v ∈ X
    for j ← 1 to |𝒮|
        if (S_j ∪ {v}) is an independent set then
            S_j ← S_j ∪ {v}
            break
        else j ← j + 1
    if j > |𝒮| then
        S_j ← {v}
        𝒮 ← 𝒮 ∪ S_j
    X ← X − {v}
```

**Figura 1:** Pseudocódigo obtenido de Lewis R. A guide to graph colouring

# Algoritmo Simulated Annealing

Función a minimizar: $H(x) = \sum_{\forall(u,v)\in E} \mathbb{1}_{\{C(u)=C(v)\}}$, con $x$ una asignación de colores y dado un parámetro $\beta$ inicial.

- Input: Grafo $\longrightarrow$ G    número de colores $\longrightarrow$ k    Iteraciones $\longrightarrow$ n
- Inicializa con un coloreo $x^0$ escogido de forma aleatoria.
- Para las siguientes iteraciones, hacer la trancisión $x^t \to x^{t+1}$ de acuerdo a los siguientes criterios:
  - Tomar vértice aleatorio $v$ con color $x_v$, tomar otro color distinto aleatorio y recolorear v. Considerando el nuevo coloreo calcular la diferencia de costo $\Delta$.
  - Si $\Delta \leq 0$, entonces se acepta el nuevo color con probabilidad 1 y $x^{t+1} = x^{new}$.
  - Si $\Delta > 0$, entonces dada una variable $U[0,1]$, si $U \leq e^{-\beta\Delta}$ se acepta el nuevo color y $x^{t+1} = x^{new}$, si ocurre lo contrario el color se rechaza y $x^{t+1} = x^t$.

# Algoritmo Simulated Annealing

- Iterar por n veces e ir aumentando $\beta$ de acuerdo a una tasa de aumento y cada cierta cantidad de iteraciones.
- Si el mínimo de la función H es 0, entonces volver a hacer lo anterior con $k-1$ colores, cuando el mínimo de H deje de ser 0, nos quedamos con el número de colores anterior y su respectiva coloración.
- Output: $x$

# Algoritmo Tabu Search

Función a minimizar: $H(S) = \sum_{\forall (u,v)\in E} \mathbb{1}_{\{C(u)=C(v)\}}$, S una particion posible.

1. Input: Grafo $\longrightarrow$ G  particion factible $\longrightarrow$ P  Iteraciones $\longrightarrow$ t
2. TabuList $\longrightarrow \emptyset$, $S_{act} \longrightarrow$ P, C* $\longrightarrow$ C(P), H* $\longrightarrow$ H(P)
3. for $I$ in range(t):
   - Ver que vértices hacen $clash$ en P  $(\alpha)$
     - si no hay vertices que hacen clash, H* $\longrightarrow$ H(S),C* $\longrightarrow$ C(S). Tomar la parte de la particion con menor grado y distribuir aleatoriamente. (reducir un color). Con P actualizado, volver a $(\alpha)$
   - Ver de todos los cambios posibles que no esten en TabuList, el cambio que produce mayor disminución de H (delta(H) $<=$ 0)
   - Si no hay mov. posibles fuera de TabuList tq delta(S) $<=$0, ver entre los mov. posibles que si esten en Tabulist que cumplan ciertos criterios y minimicen H(S´), realizarlo.
   - Actualizar TabuList, Actualizar $S_{act}$
4. Output: C(S)

# Resultados Obtenidos

**Tabla 1:** Cantidad media de colores aproximada entregados por Tabu Search y Simulated Annealing para un tiempo de simulación determinado y un grafo del tipo Erdos-Renyi de 1000 vértices con grado promedio de vértice 10 comenzando con 100 colores.

| | **30[s]** | **60[s]** |
|---|---|---|
| **Tabu Search** | 7 | 6 |
| **S. Annealing** | 92 | 88 |

# Resultados Obtenidos

**Tabla 2:** Cantidad media de colores proximada entregados por Tabu Search y Simulated Annealing para un tiempo de simulación determinado y un grafo del tipo Erdos-Renyi de 1000 vértices con grado promedio de vértice 5 comenzando con 100 colores.

| | **30[s]** | **60[s]** |
|---|---|---|
| **Tabu Search** | 12 | 12 |
| **S. Annealing** | 91 | 87 |

# Resultados

**Tabla 3:** Cantidad de colores en media entregados por los 3 algoritmos para distintos tipos de grafos.

| Tipo de grafo | DSatur | S. Annealing | TabuSearch |
|---|---|---|---|
| **Bipartito(50,50,0.1)** | 2 | 38 | 3 |
| **Erdos-Renyi(1000,0.005)** | 8 | 4 | 5 |
| **Completo(40)** | 40 | 40 | 40 |

# Referencias


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/bl
4. Salhi S.(2000). Defining tabu list size and aspiration criterion within tabu search methods.

# Proyecto final Simulación Estocástica

## Simulated Annealing aplicado al coloreamiento de grafos

Rodrigo Altamirano M., Gonzalo Ovalle J.

Universidad de Chile
Facultad de Ciencias Físicas y Matemáticas
Departamento de Ingeniería Matemática

21 de diciembre de 2022