

# **Práctica 5: “Árbol de Máximo y Mínimo coste Kruskal”**

**Angel Bernardo Márquez Valdivia**

**22110348**

**Inteligencia Artificial**

**6E2**



## ¿Qué es?

El Árbol de Máximo y Mínimo Costo basado en Kruskal es una extensión del algoritmo clásico de Kruskal, utilizado en la teoría de grafos para encontrar un árbol generador mínimo o máximo de un grafo conexo y ponderado.

- **Árbol Generador**

Un árbol generador de un grafo es un subgrafo que conecta todos los vértices del grafo original con el menor número de aristas posible ( $n-1$  aristas, donde  $n$  es el número de vértices) y sin formar ciclos.

- **Árbol Generador Mínimo (MST)**

Es un árbol generador en el que la suma de los pesos de sus aristas es la mínima posible.

- **Árbol Generador Máximo**

Es un árbol generador en el que la suma de los pesos de sus aristas es la máxima posible.

- **Algoritmo de Kruskal**

Es un algoritmo utilizado para encontrar el Árbol Generador Mínimo de un grafo. También puede modificarse para encontrar el Árbol Generador Máximo invirtiendo el orden de selección de las aristas.

## Características del Árbol de Máximo y Mínimo Costo

### 1. Propiedades Generales

- **Conectividad:** El árbol conecta todos los vértices del grafo original.
- **Peso total:** En un Árbol de Mínimo Costo, el peso total es mínimo. En un Árbol de Máximo Costo, es el máximo.
- **Estructura acíclica:** Por definición, un árbol generador no contiene ciclos.
- **Optimización:** Ambos buscan optimizar (minimizar o maximizar) el peso total de las aristas seleccionadas.

### 2. Propiedades Específicas

- **Ordenación de aristas:** El Árbol de Máximo Costo invierte el orden de las aristas, seleccionando primero las más pesadas.
- **Complejidad computacional:** Ambos árboles tienen la misma complejidad, ya que solo cambia el criterio de ordenación de las aristas.

### **3. Algoritmo Base: Kruskal**

- El algoritmo de Kruskal es un enfoque greedy (voraz) que selecciona la mejor opción en cada paso con la esperanza de alcanzar la solución óptima global.
- Funciona ordenando las aristas según su peso y añadiéndolas progresivamente al árbol, siempre que no formen ciclos.

## **Funcionamiento del Algoritmo de Kruskal**

### **Pasos Generales**

#### **1. Entrada del grafo:**

Representación del grafo como una lista de aristas, donde cada arista tiene un peso asociado.

#### **2. Ordenar las aristas:**

Para el Árbol de Mínimo Costo, las aristas se ordenan de forma ascendente según su peso.

Para el Árbol de Máximo Costo, las aristas se ordenan de forma descendente.

#### **3. Inicializar conjuntos disjuntos:**

Cada vértice del grafo se trata como un conjunto independiente (estructura Union-Find).

#### **4. Construcción del árbol:**

Iterar sobre la lista de aristas en el orden determinado.

Añadir una arista al árbol si conecta dos componentes diferentes (no forma ciclos).

Unir los conjuntos de los vértices conectados por la arista.

#### **5. Finalización:**

Detenerse cuando el árbol contenga  $n - 1$  aristas, donde  $n$  es el número de vértices del grafo.

## **¿Para Qué Sirve el Árbol de Máximo y Mínimo Costo?**

### **Árbol de Mínimo Costo**

#### **• Diseño de Redes:**

Minimización del costo al construir redes eléctricas, de transporte, o de telecomunicaciones.

- **Rutas óptimas:**

Encontrar rutas de menor costo en sistemas de distribución y logística.

### **Árbol de Máximo Costo**

- **Análisis de Conexiones Críticas:**

Identificar las conexiones más importantes (o costosas) en un sistema.

- **Robustez en redes:**

Diseñar redes donde las conexiones más fuertes o de mayor capacidad se prioricen.

- **Sistemas de Seguridad:**

Maximización de rutas seguras en redes vulnerables.

## **Aplicaciones Reales**

- **Telecomunicaciones:**

Minimización de costos en la construcción de redes de fibra óptica.

- **Sistemas de Transporte:**

Planificación de carreteras y rutas óptimas para sistemas logísticos.

- **Sistemas de Energía:**

Diseño de redes eléctricas eficientes.

- **Biología Computacional:**

Análisis de estructuras moleculares o redes biológicas donde las conexiones fuertes son críticas.

## **¿Cómo se implementa en el mundo?**

El Árbol de Máximo y Mínimo Costo basado en el algoritmo de Kruskal tiene diversas aplicaciones prácticas en múltiples disciplinas, desde la ingeniería hasta la biología y las ciencias de datos. Su implementación en el mundo real varía según los objetivos específicos del sistema o problema que se esté abordando. A continuación, exploramos cómo se implementa en diferentes áreas del mundo:

## **Implementación del Árbol de Mínimo Costo**

El Árbol de Mínimo Costo se usa cuando el objetivo es optimizar recursos, minimizar costos, o conectar componentes con la mayor eficiencia posible. Algunas de sus aplicaciones e implementaciones clave incluyen:

### **1. Redes de Telecomunicaciones**

Problema: Diseñar una red de telecomunicaciones (fibra óptica, cables, antenas) que conecte ciudades o nodos con el menor costo.

#### **Implementación:**

Los nodos representan ciudades, y las aristas representan posibles conexiones entre ellas con un costo asociado.

El algoritmo de Kruskal se utiliza para seleccionar las conexiones más baratas hasta conectar todas las ciudades sin redundancia.

Ejemplo real: Implementado en el diseño de redes de fibra óptica para grandes empresas como Google o proveedores de Internet.

### **2. Redes Eléctricas**

Problema: Construir redes de distribución eléctrica con la menor longitud total de cables.

#### **Implementación:**

Cada subestación se representa como un nodo, y las posibles conexiones como aristas ponderadas por la distancia o costo.

El Árbol de Mínimo Costo asegura la conexión entre todas las subestaciones minimizando el material necesario.

Ejemplo real: En proyectos de electrificación rural o distribución urbana de energía.

### **3. Sistemas de Transporte**

Problema: Diseñar una red de carreteras o ferrocarriles que minimice los costos de construcción.

#### **Implementación:**

Los nodos representan puntos importantes (ciudades, estaciones, etc.), y las aristas son las posibles conexiones con costos de construcción.

El Árbol de Mínimo Costo se aplica para garantizar una red eficiente.

Ejemplo real: La planificación de rutas ferroviarias de bajo costo en países en desarrollo.

#### **4. Ciencia de Datos y Clustering**

Problema: Agrupar datos de manera eficiente utilizando modelos jerárquicos.

##### **Implementación:**

Los puntos de datos se consideran nodos, y las similitudes entre ellos se representan como aristas con un peso inversamente proporcional a la similitud.

Un Árbol de Mínimo Costo conecta todos los datos, optimizando la agrupación jerárquica.

Ejemplo real: En aprendizaje automático y análisis de grandes datos.

#### **Implementación del Árbol de Máximo Costo**

El Árbol de Máximo Costo se utiliza cuando el objetivo es maximizar la robustez, la capacidad, o seleccionar las conexiones más fuertes. Algunas aplicaciones incluyen:

##### **1. Análisis de Redes Sociales**

Problema: Identificar las conexiones más influyentes o críticas en una red social.

##### **Implementación:**

Los nodos representan usuarios, y las aristas representan interacciones con un peso basado en la fuerza de la interacción.

El Árbol de Máximo Costo selecciona las relaciones más influyentes para análisis.

Ejemplo real: Identificar líderes de opinión o comunidades clave en plataformas como Twitter o Facebook.

##### **2. Diseño de Redes Resilientes**

Problema: Diseñar redes de comunicación que sean altamente robustas frente a fallos.

##### **Implementación:**

Se priorizan las conexiones más fuertes o de mayor capacidad.

El Árbol de Máximo Costo asegura que las conexiones críticas estén representadas, maximizando la resiliencia de la red.

Ejemplo real: Redes de defensa y comunicación militar.

##### **3. Gestión de Infraestructuras**

Problema: Garantizar la máxima utilización de recursos críticos en sistemas de infraestructura.

### **Implementación:**

Los recursos se modelan como nodos y las capacidades máximas como pesos en las aristas.

El Árbol de Máximo Costo selecciona las rutas de mayor capacidad.

Ejemplo real: Planificación de sistemas hidráulicos o de gasoductos.

## **4. Biología Computacional**

Problema: Analizar redes moleculares o filogenéticas priorizando conexiones importantes.

### **Implementación:**

Los nodos representan moléculas o especies, y las aristas representan relaciones con pesos que indican importancia.

El Árbol de Máximo Costo prioriza las interacciones más relevantes para el análisis.

Ejemplo real: Estudios de redes metabólicas o reconstrucción de árboles evolutivos.

## **¿Cómo lo implementarías en tu vida?**

En mi día a día, considero que el Árbol de Máximo y Mínimo Costo de Kruskal es una herramienta muy útil para optimizar diversas áreas de mi vida. A continuación, comparto algunos ejemplos de cómo podría aplicarlo en tareas, proyectos, viajes y en mi economía personal.

### **Organización de Tareas y Proyectos**

Cuando tengo muchas tareas o proyectos por realizar, suelo sentirme abrumado tratando de decidir por dónde empezar. Aquí es donde el Árbol de Mínimo Costo puede ayudarme. Cada tarea la considero como un nodo, y las dependencias o esfuerzos necesarios para completarla son las aristas con un peso asociado. Usando este enfoque, puedo priorizar las tareas que requieren menos tiempo o esfuerzo, logrando así completar mis proyectos de manera eficiente.

### **Planificación de Rutas de Viaje**

Cuando planeo viajar, siempre busco la forma más eficiente y económica de llegar a mi destino mientras aprovecho al máximo mi tiempo. Aquí, los destinos serían los nodos y las rutas entre ellos, las aristas con un costo asignado (ya sea tiempo, dinero o distancia). Usando el Árbol de Mínimo Costo, puedo identificar la mejor ruta que minimice estos factores. Esto me permite ahorrar recursos y disfrutar más de mi experiencia de viaje.

## **Optimización de Gastos Mensuales**

Gestionar mi economía personal es una de mis prioridades, y aplicar el Árbol de Mínimo Costo me resulta muy útil. Por ejemplo, puedo considerar diferentes aspectos de mis gastos mensuales (como alimentación, transporte o entretenimiento) como nodos, y las opciones para cubrir cada uno como aristas con pesos que reflejan el costo asociado. Con este enfoque, puedo seleccionar las alternativas que me permitan satisfacer mis necesidades mientras minimizo mis gastos totales. De esta forma, logro mantener un equilibrio financiero y aprovechar mejor mi presupuesto.

## **¿Cómo lo implementarías en tu trabajo o tu trabajo de ensueño?**

Trabajar en una empresa dedicada a la robótica y automatización, especialmente en el área de vehículos, sería para mí un sueño hecho realidad. Estoy convencido de que los principios del Árbol de Máximo y Mínimo Costo de Kruskal pueden ser una herramienta valiosa en este tipo de entorno, ya que permiten optimizar procesos, reducir costos y tomar decisiones más inteligentes. Aquí comparto algunas ideas sobre cómo implementaría este concepto en mi día a día laboral:

- **Optimización en los Sensores en Vehículos Autónomos**

En el diseño y desarrollo de vehículos autónomos, una de las principales tareas es optimizar la comunicación entre los sensores. Cada sensor puede considerarse como un nodo, y las conexiones entre ellos (como redes de datos o cables) serían las aristas con un peso asociado, como el costo, la latencia o la eficiencia energética. Utilizando el Árbol de Mínimo Costo, podría identificar la configuración más eficiente para garantizar una comunicación rápida y estable entre los sensores, minimizando el uso de recursos.

- **Planificación de Rutas para Vehículos Autónomos**

El Árbol de Mínimo Costo sería crucial en la planificación de rutas de los vehículos. Por ejemplo, al determinar el trayecto más eficiente para un vehículo de reparto, podría considerar los destinos como nodos y las rutas entre ellos como aristas, asignando pesos que representen factores como la distancia, el tiempo de viaje o el consumo de energía. De esta forma, sería posible optimizar las rutas y garantizar que los vehículos lleguen a sus destinos de manera eficiente.



- **Gestión de Proyectos de Automatización**

En la industria de la robótica, los proyectos suelen ser complejos y requieren priorización de tareas. Aquí, usaría el Árbol de Mínimo Costo para identificar la secuencia de tareas que minimice el tiempo y los recursos necesarios para completar un proyecto, asegurándome de que el equipo trabaje de manera eficiente.

- **Optimización de Sistemas de Energía en Vehículos**

En los vehículos autónomos, la gestión energética es un factor crítico. Podría usar el Árbol de Mínimo Costo para optimizar la distribución de energía en los sistemas del vehículo, asegurando que los componentes clave (como motores, sensores y unidades de procesamiento) reciban la energía necesaria con el menor desperdicio posible.

## **Ejemplo del Árbol de Máximo y Mínimo coste Kruskal.**

El árbol de mínimo costo es una estructura que conecta todos los nodos de un grafo de manera que la suma total de las aristas es mínima, sin formar ciclos. Kruskal es uno de los algoritmos utilizados para esto.

El árbol de máximo costo es el opuesto: maximiza la suma total de las aristas mientras sigue evitando ciclos.

### **Funcionamiento del Código**

El código es una aplicación con una interfaz gráfica (Tkinter) que permite al usuario:

- Agregar nodos y aristas con sus pesos.
- Calcular el árbol mínimo o máximo basado en el algoritmo de Kruskal.
- Visualizar la representación gráfica de los nodos y las aristas en el canvas.

## **Interfaz Gráfica**

### **1. Componentes Principales**

El código utiliza la librería tkinter para crear una interfaz amigable:

- **Entrada de datos:**

El usuario introduce nodos (Nodo 1, Nodo 2) y el peso de las aristas en el área de entrada.

Al hacer clic en el botón "Agregar Arista", la arista se agrega a la lista de aristas y se actualiza la representación gráfica.

- **Selección del tipo de árbol:**

El usuario elige si desea calcular el "Mínimo Costo" o el "Máximo Costo" de las aristas usando un `ttk.Combobox`.

- **Cálculo del árbol:**

Al hacer clic en el botón "Calcular Árbol", se ejecuta el algoritmo de Kruskal para construir el Árbol Generador Mínimo (MST) o Máximo según la selección del usuario.

## Componentes del Código

- **Clase Principal: KruskalGUI**

Esta clase encapsula toda la lógica, la creación de widgets, la interacción con el usuario y el cálculo de Kruskal.

1. **`self.edges` y `self.nodes`**

- **`self.edges`:** Lista para almacenar las aristas con su peso (peso, nodo1, nodo2).
- **`self.nodes`:** Conjunto para rastrear todos los nodos que se han ingresado.

2. **`self.create_widgets()`**

Crea los elementos visuales en la ventana:

- Entrada para nodos y peso.
- Botón para agregar aristas.
- `Combobox` para seleccionar el tipo de cálculo (Mínimo Costo o Máximo Costo).
- Botón para ejecutar el cálculo.
- Área de texto para mostrar los resultados.
- `Canvas` para visualizar la estructura gráfica.

## Algoritmo de Kruskal (pasos)

1. Ordenar todas las aristas por peso (de menor a mayor para mínimo costo, o mayor a menor para máximo costo).
2. Inicializar cada nodo como un conjunto separado.
3. Iterar sobre las aristas ordenadas:
  - Si los nodos de una arista no están conectados, incluir esa arista en el árbol y unir sus conjuntos.
4. Detenerse cuando todos los nodos estén conectados en un árbol sin ciclos.

Árbol de Máximo y Mínimo Costo - Kruskal

Nodo 1:  Nodo 2:  Peso:

Selecciona tipo de Árbol:

```
Arista agregada: B - C con peso 21
Arista agregada: A - C con peso 2
Arista agregada: C - D con peso 5
Arista agregada: D - E con peso 8
Arista agregada: B - D con peso 8
Arista agregada: A - D con peso 5
Iniciando el cálculo del Árbol...
Arista añadida al árbol: A - C con peso 2
Arista añadida al árbol: A - D con peso 5
Arista añadida al árbol: B - D con peso 8
Arista añadida al árbol: D - E con peso 8

Árbol generado:
A - C con peso 2
A - D con peso 5
B - D con peso 8
D - E con peso 8

Costo total del Árbol: 23
```

