



TECNOLÓGICO  
NACIONAL DE MÉXICO



# Instituto tecnológico de Culiacán

## **Actividad:**

Problemas de optimización y algoritmos

## **Alumno:**

Ivan Eduardo Ramírez moreno

## **Docente:**

ZURIEL DATHAN MORA FELIX

## **Materia:**

Tópicos de IA

## **Numero de control:**

20170787

## **Semestre:**

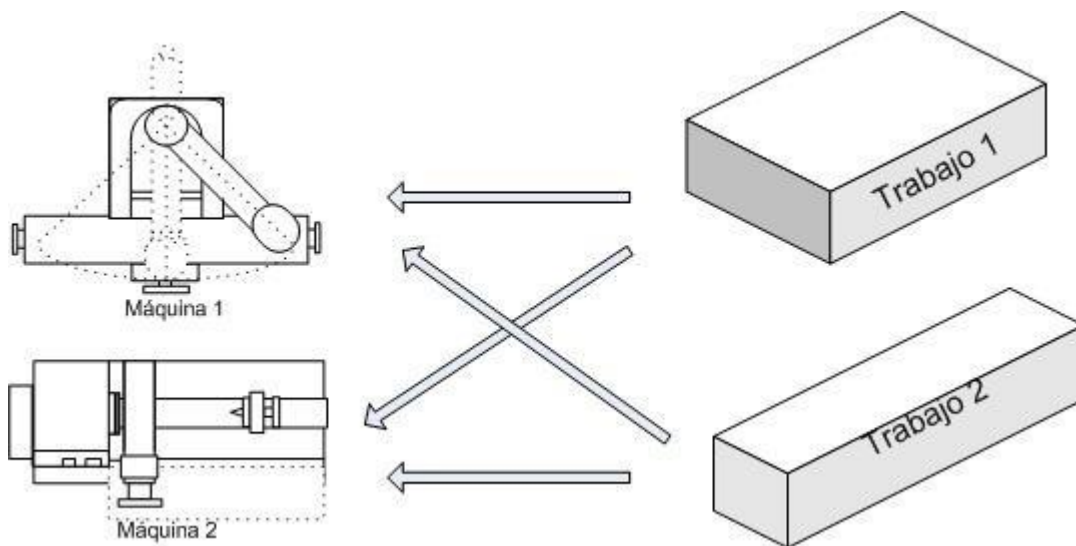
10

## 1. Problema de Programación de Trabajos (JSSP - Job Shop Scheduling Problem)

### Descripción:

El **problema de programación de trabajos (JSSP)** es un problema de optimización combinatoria en el que se busca programar un conjunto de trabajos en varias máquinas con restricciones específicas. Cada trabajo consta de una secuencia de operaciones que deben realizarse en un orden específico y en diferentes máquinas. El objetivo es minimizar el tiempo total de finalización (**makespan**) o mejorar la eficiencia en función de otros criterios, como tiempos de espera o utilización de recursos.

Es un problema **NP-difícil**, lo que significa que no existe un algoritmo eficiente para encontrar la solución óptima en todos los casos. Se utilizan heurísticas y algoritmos metaheurísticos como Branch and Bound, Algoritmos Genéticos, y Simulated Annealing para abordarlo.



### Ejemplo:

Supongamos que tenemos **3 trabajos (J1, J2, J3)** y **3 máquinas (M1, M2, M3)**, con los siguientes tiempos de procesamiento:

Trabajo	M1	M2	M3
J1	3	2	2
J2	2	1	4
J3	4	3	1

Cada trabajo debe ejecutarse en el **orden indicado** y no puede ser interrumpido. El objetivo es minimizar el **makespan** (tiempo total de finalización).

### Métodos de resolución:

- Algoritmo de Branch and Bound (explora todas las combinaciones posibles, pero es ineficiente para grandes N y M).
- Algoritmos heurísticos como Simulated Annealing o Algoritmos Genéticos.
- Despacho de reglas (Ejemplo: Primero el trabajo más corto, Mayor prioridad a trabajos críticos).

### Aplicaciones:

- Planificación en fábricas y manufactura.
- Optimización en centros de datos y servidores.
- Logística y distribución para minimizar tiempos de espera.

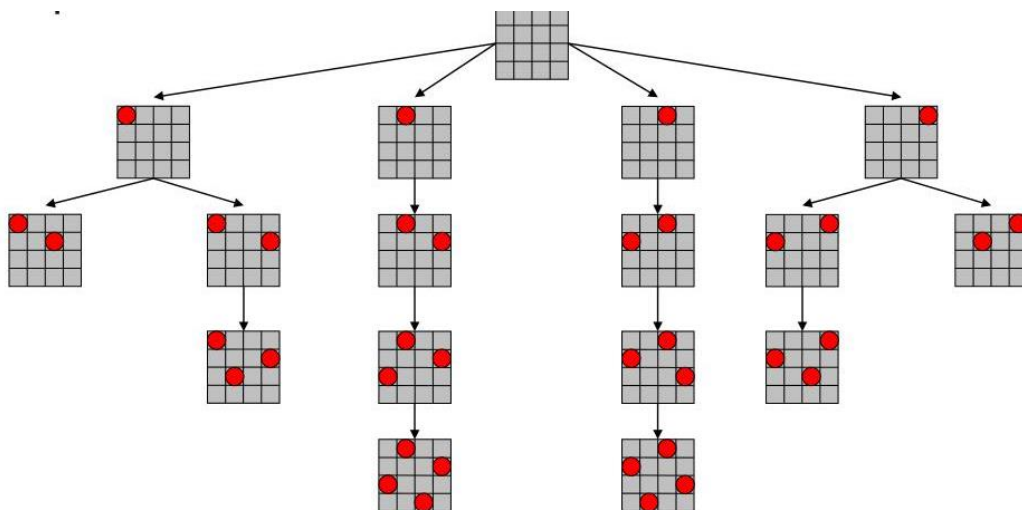
## 2. Problema de las N reinas

### Descripción:

El **problema de las N reinas** consiste en colocar **N reinas** en un tablero de ajedrez de **N** × **N** de tal manera que ninguna reina ataque a otra. Esto significa que no pueden estar en la misma fila, columna o diagonal.

El problema puede resolverse usando **backtracking**, técnicas de búsqueda heurística como **algoritmos genéticos**, o técnicas de **programación lineal**.

Es un problema clásico de inteligencia artificial y optimización combinatoria, con aplicaciones en criptografía, redes y sistemas distribuidos.



## Métodos de resolución:

### 1. Backtracking:

- Colocar una reina en una columna, verificar si es válida.
- Si hay conflicto, retroceder (**backtrack**) y probar otra posición.

### 2. Algoritmos heurísticos:

- **Hill Climbing**: Ajusta la posición de las reinas iterativamente.
- **Simulated Annealing**: Reduce conflictos de manera probabilística.

## Aplicaciones:

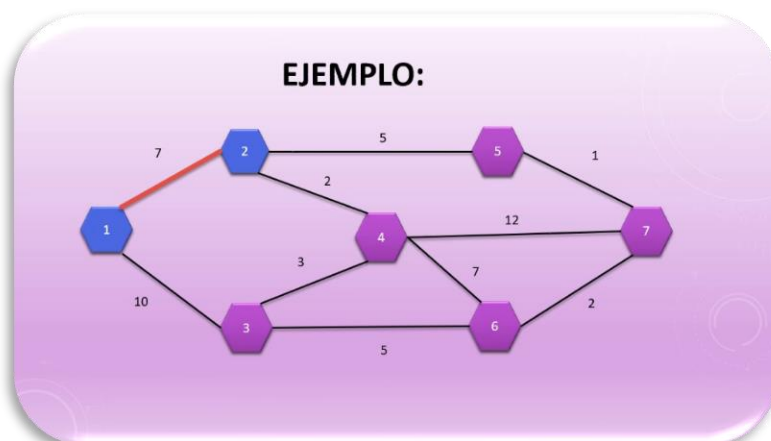
- En **inteligencia artificial**, este problema ayuda a modelar problemas de búsqueda y optimización.
- Se usa en **criptografía** y diseño de redes para evitar colisiones en canales de comunicación.

## 3. Árbol de Expansión Mínima (MST - Minimum Spanning Tree)

### Descripción:

Dado un grafo ponderado **conectado y no dirigido**, un **árbol de expansión mínima (MST)** es un subconjunto de las aristas que conecta todos los vértices sin formar ciclos y con la suma mínima de pesos de las aristas.

El MST es útil en problemas de **redes de comunicación**, planificación de circuitos eléctricos, diseño de redes de transporte y clustering en aprendizaje automático.



### Algoritmos más utilizados:

1. **Algoritmo de Kruskal** (selecciona primero las aristas más livianas).
2. **Algoritmo de Prim** (expande un árbol desde un nodo inicial).

### Aplicaciones:

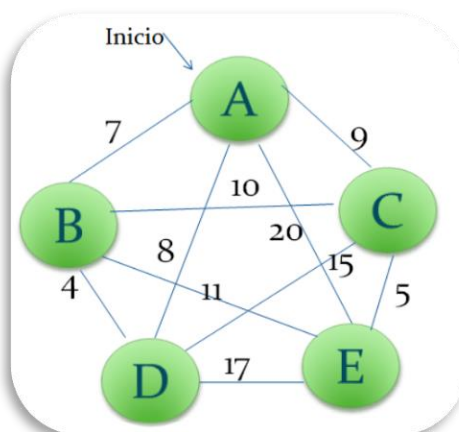
- **Diseño de redes de telecomunicaciones** (fibra óptica, 5G).
- **Planificación de carreteras** y redes de distribución eléctrica.

## 4. Problema del Agente Viajero (TSP - Traveling Salesman Problem)

### Descripción:

El **problema del agente viajero (TSP)** es un problema de optimización en el que un viajero debe visitar un conjunto de **n ciudades**, pasando una sola vez por cada una y regresando a la ciudad inicial, minimizando la distancia total recorrida.

Es un problema **NP-difícil**, ampliamente estudiado en inteligencia artificial y teoría de la computación. Se usa en logística, planificación de rutas, manufactura y diseño de circuitos.



### Ejemplo:

Dado un conjunto de ciudades y distancias:

	A	B	C	D
A	0	10	15	20
B	10	0	35	25
C	15	35	0	30
D	20	25	30	0

Una posible solución sería: **A → B → D → C → A** con distancia total **80**.

### **Algoritmos de resolución:**

- **Fuerza bruta:** Prueba todas las combinaciones posibles ( **$N!$**  soluciones).
- Algoritmo de Aproximación de Vecino Más Cercano.
- **Algoritmos evolutivos:** Algoritmos Genéticos, Colonia de Hormigas.

### **Aplicaciones:**

- **Optimización de rutas de entrega** (Amazon, FedEx).
- **Diseño de circuitos eléctricos** (minimización de cableado).
- **Secuenciación del ADN** (problemas de alineación de genes).

### **Conclusión:**

Estos cuatro problemas son fundamentales en optimización combinatoria y tienen aplicaciones reales en la industria, telecomunicaciones, inteligencia artificial y logística. Si necesitas ayuda con código para resolver alguno de ellos, dime y te ayudo a programarlo en Python o C++.

## Referencias:

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to algorithms* (3rd ed.). The MIT Press.

Garey, M. R., & Johnson, D. S. (1979). *Computers and intractability: A guide to the theory of NP-completeness*. W. H. Freeman.

Kleinberg, J., & Tardos, É. (2005). *Algorithm design*. Pearson Education.

Papadimitriou, C. H., & Steiglitz, K. (1998). *Combinatorial optimization: Algorithms and complexity*. Dover Publications.

Sedgewick, R., & Wayne, K. (2011). *Algorithms* (4th ed.). Addison-Wesley.

Tarjan, R. E. (1983). *Data structures and network algorithms*. SIAM.

Wikipedia contributors. (2024). *Job shop scheduling*. Wikipedia, The Free Encyclopedia. Retrieved March 7, 2025, from [https://en.wikipedia.org/wiki/Job\\_shop\\_scheduling](https://en.wikipedia.org/wiki/Job_shop_scheduling)

Wikipedia contributors. (2024). *N-queens problem*. Wikipedia, The Free Encyclopedia. Retrieved March 7, 2025, from <https://en.wikipedia.org/wiki/N-queens>

Wikipedia contributors. (2024). *Minimum spanning tree*. Wikipedia, The Free Encyclopedia. Retrieved March 7, 2025, from [https://en.wikipedia.org/wiki/Minimum\\_spanning\\_tree](https://en.wikipedia.org/wiki/Minimum_spanning_tree)

Wikipedia contributors. (2024). *Travelling salesman problem*. Wikipedia, The Free Encyclopedia. Retrieved March 7, 2025, from [https://en.wikipedia.org/wiki/Travelling\\_salesman\\_problem](https://en.wikipedia.org/wiki/Travelling_salesman_problem)