INSTITUTO TECNOLÓGICO DE CULIACÁN

Materia:

Tópicos de IA

Tarea#1 Unidad 2

Profesor:

Zuriel Dathan Mora Felix

Alumna:

Anette Leticia Robles Zamora

1- Problema de programación de trabajos (JSSP)

El Problema de Programación de Trabajos (JSSP, por sus siglas en inglés: Job Shop Scheduling Problem) es un problema clásico de optimización combinatoria en el ámbito de la planificación y asignación de recursos en sistemas de producción. Es considerado NP-difícil, lo que significa que encontrar una solución óptima para instancias grandes es computacionalmente complejo

El JSSP consiste en programar n trabajos en m máquinas, donde:

- Cada trabajo tiene una secuencia específica de operaciones que deben realizarse en ciertas máquinas.
- Cada máquina solo puede procesar una operación a la vez.
- Una vez que una operación comienza en una máquina, debe completarse sin interrupciones.
- El objetivo es optimizar un criterio, típicamente **minimizar el tiempo total de finalización de los trabajos** (*makespan*, denotado como **Cmax**).

Métodos de Resolución

1. Métodos Exactos:

- **Programación Lineal Entera (ILP)**: Modela el problema con restricciones matemáticas y lo resuelve con técnicas como el **Branch and Bound**.
- **Enumeración Exhaustiva**: Explora todas las posibles secuencias, pero es computacionalmente inviable para grandes instancias.

2. Métodos Heurísticos:

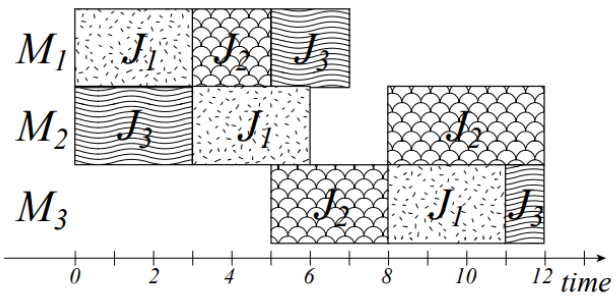
- **Reglas de Despacho** (Dispatching Rules): Algoritmos simples como **SPT (Shortest Processing Time)**, **LPT (Longest Processing Time)**, **EDD (Earliest Due Date)**.
- **Algoritmos Constructivos**: Métodos como **Greedy** que generan soluciones factibles rápidamente.

3. Métodos Metaheurísticos:

- **Algoritmos Genéticos (GA)**: Evolucionan soluciones mediante selección, mutación y cruce.
- **Recocido Simulado (Simulated Annealing - SA)**: Explora el espacio de soluciones permitiendo ocasionalmente movimientos subóptimos para evitar óptimos locales.

- **Optimización por Colonia de Hormigas (ACO)**: Basado en el comportamiento de hormigas para encontrar rutas óptimas.

Se representa de la siguiente manera:



- J1-J3 representa los trabajos
- M1-M3 significa las Máquinas
- La longitud de esta solución es 12, que es la primera vez que se completan los tres trabajos. Sin embargo, tenga en cuenta que esta no es la solución óptima.

2- Problema de las N reynas

El problema de las N-Reinas consiste en colocar n reinas en un tablero de ajedrez de tamaño N*N de forma la reinas no se amenacen según las normas del ajedrez. Se busca encontrar una solución o todas las soluciones posibles.

El objetivo es encontrar una configuración válida donde todas las reinas estén colocadas sin conflictos.

Métodos de Resolución

1. Búsqueda Exhaustiva (Fuerza Bruta)

- Explora todas las combinaciones posibles de ubicación de las reinas.
- Para cada disposición, verifica si cumple las restricciones.
- Tiene una complejidad de **O(N!)**, lo que lo hace ineficiente para valores grandes de **N**.

2. Backtracking (Vuelta Atrás)

- Construye una solución paso a paso, retrocediendo cuando se detecta una violación de restricciones.

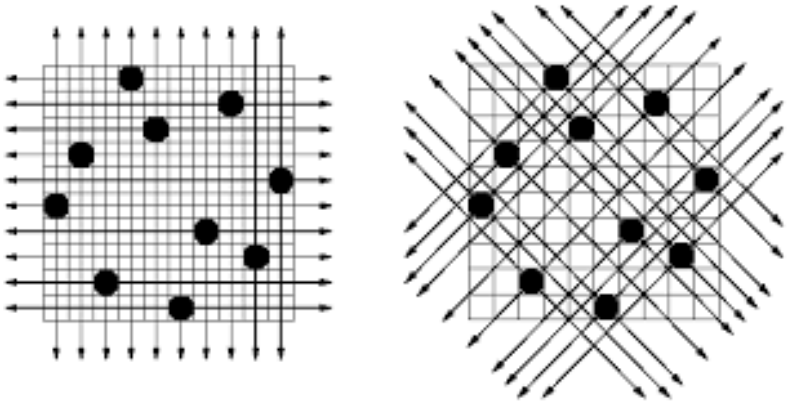
- Mejora el tiempo de ejecución en comparación con la fuerza bruta.
- Tiene una complejidad en el peor caso de **O(N!)**, pero funciona mejor en la práctica.

3. Algoritmos Metaheurísticos

Cuando **N** es grande, los métodos exactos son poco eficientes, por lo que se usan algoritmos aproximados:

- **Algoritmos Genéticos**: Evolucionan poblaciones de soluciones usando selección, mutación y cruce.
- **Recocido Simulado (Simulated Annealing)**: Explora soluciones probables y permite cambios temporales hacia soluciones peores para evitar mínimos locales.
- **Optimización por Enjambre de Partículas (PSO)**: Técnica inspirada en la inteligencia de enjambres para encontrar soluciones óptimas.

Se representa de la siguiente manera:



3- Árbol de expansión mínima (MST)

Un árbol de expansión mínima (MST) se define como un árbol de longitud total mínima que conecta un conjunto de puntos, donde los nodos representan los puntos y los bordes son segmentos de línea que unen pares de puntos.

El **MST** es único si todos los pesos de las aristas son distintos, pero si hay pesos repetidos, pueden existir múltiples árboles de expansión mínima con el mismo costo.

Su objetivo es encontrar un subconjunto de las aristas de un grafo ponderado y conexo que conecte todos los vértices del grafo sin formar ciclos, de manera que la suma total de los pesos de las aristas seleccionadas sea mínima.

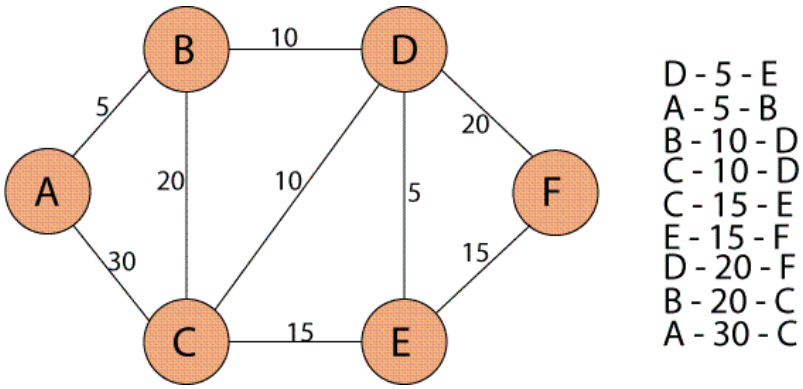
Métodos de Resolución

1.Kruskal: Ordena todas las aristas por peso y las añade al MST en orden ascendente, evitando ciclos mediante un Disjoint Set Union (DSU). Su complejidad es O(E log E), lo que lo hace eficiente en grafos dispersos.

2.Prim: Comienza desde un nodo arbitrario y expande el MST agregando la arista más barata que conecta un nodo dentro del MST con uno fuera de él. Usando una cola de prioridad, su complejidad es O(E log V), siendo más eficiente en grafos densos.

3.Borůvka: Selecciona iterativamente la arista de menor peso para cada componente conectado, fusionando componentes hasta obtener un solo MST. Su complejidad es O(E log V) y es útil en implementaciones paralelas.

Se representa de la siguiente manera:



4- Problema del agente viajero (TSP)

El Problema del Agente Viajero (TSP, por sus siglas en inglés) es uno de los problemas de optimización más estudiados en la ciencia de la computación. El problema consiste en encontrar la ruta más corta para un vendedor que visite una lista de ciudades

exactamente una vez y regrese a la ciudad de origen. Normalmente, se representa el esquema de ciudades como un grafo completo, donde cada arista tiene un peso asociado que representa la distancia entre ciudades. El TSP se relaciona con los ciclos hamiltonianos, que son caminos cíclicos en grafos que pasan por cada nodo una única vez y tienen el mismo punto de llegada y salida.

Métodos Exactos:

1.Fuerza Bruta: Evalúa todas las permutaciones posibles de las ciudades, con una complejidad O(N!), lo que lo hace ineficiente para muchos puntos.

- Programación Dinámica: El algoritmo de Held-Karp tiene una complejidad de O(N^2 * 2^N), mejorando la fuerza bruta, pero aún es costoso para grandes instancias.
- Branch and Bound: Utiliza poda para reducir el número de soluciones evaluadas, aunque sigue teniendo una complejidad exponencial en el peor caso.

2.Métodos Aproximados y Heurísticos:

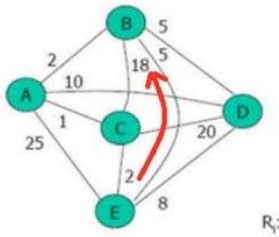
- Vecino Más Cercano: Empieza en una ciudad y elige la ciudad más cercana, rápido pero no siempre óptimo.
- Inserción más Barata: Inserta la ciudad en el lugar más barato de una solución parcial, obteniendo una buena aproximación.
- Recocido Simulado: Usa movimientos hacia soluciones peores para evitar óptimos locales, buscando una solución mejor en un espacio amplio.
- Algoritmos Genéticos: Basados en selección, cruce y mutación de soluciones, inspirados en la evolución natural.
- Optimización por Enjambre de Partículas (PSO): Emula el comportamiento de los enjambres para explorar soluciones distribuidas.

3.Algoritmos Aproximados con Garantías:

Algoritmo de Christofides: Ofrece una solución dentro de un 1.5 veces del costo óptimo, solo en grafos con distancias que cumplen la propiedad de triangularidad.

Se representa de la siguiente manera:

Estando ahora en la cuidad E, elegimos la cuidad más cercada desde allí, sería entonces B.



	B	C	D	E
A	2	1	10	25
B		18	5	5
C			20	2
D				8

R_i: Visitar ciudad i, i = A, B, C, D, E

$i = \{A, C, E, A\}$

Referencias:

https://medium.datadriveninvestor.com/job-shop-scheduling-problem-jssp-an-overview-cd99970a02f8

https://docencia.eafranco.com/materiales/estructurasdedatos/practicas/05/Practica05.pdf

https://www.sciencedirect.com/topics/computer-science/minimum-spanning-tree

https://repositorio.uniandes.edu.co/entities/publication/0acfa8ef-7378-4821-aac9-2042e3462d1c