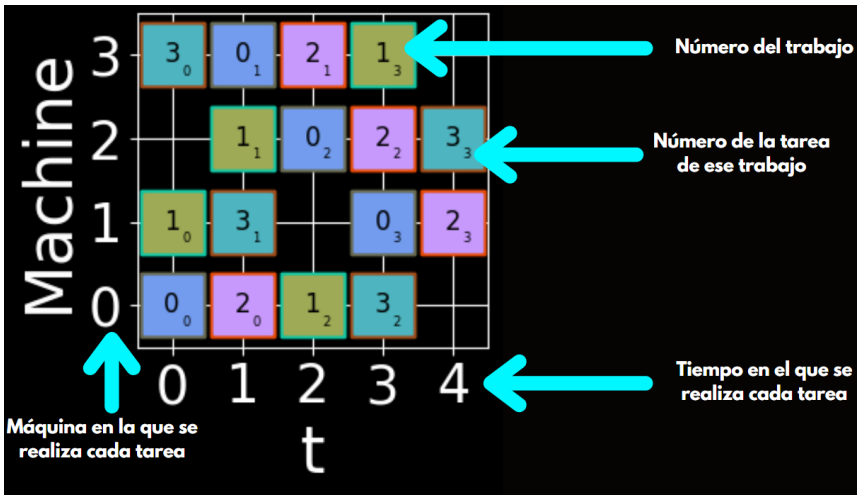
Job Shop Scheduling Problem

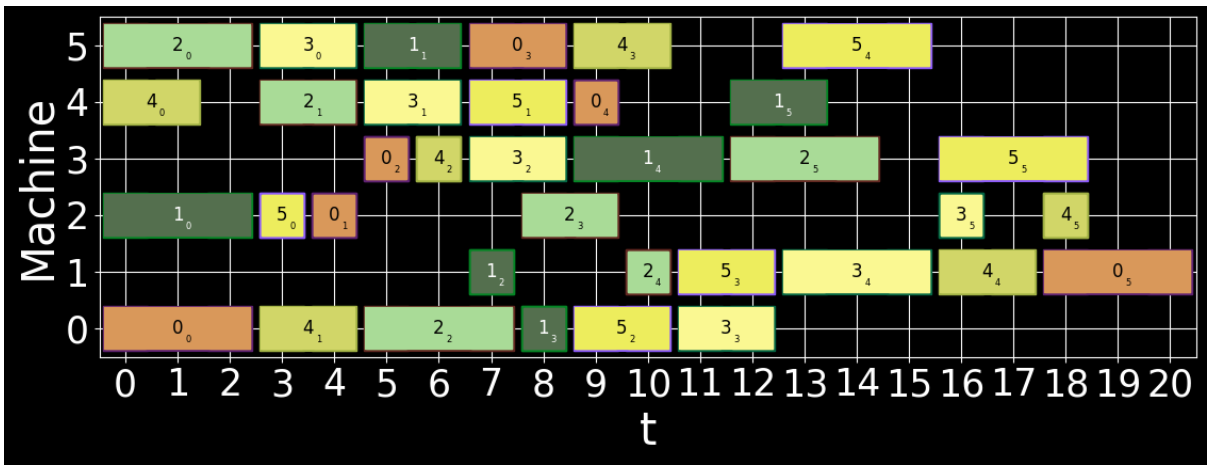
El Job Shop Scheduling Problem (JSSP) es un problema de optimización combinatoria con especial aplicabilidad en entornos industriales y altamente complejo para casos medianos y grandes.

Este consiste en la planificación de una serie de trabajos, compuesto cada uno de ellos por un conjunto de tareas, en un conjunto de máquinas, de manera que el tiempo de ejecución total sea el mínimo posible. Cada tarea puede ejecutarse únicamente en una máquina concreta y tiene una duración concreta, las tareas de cada trabajo se tienen que realizar en un orden concreto, y cada máquina solo puede realizar una tarea a la vez.

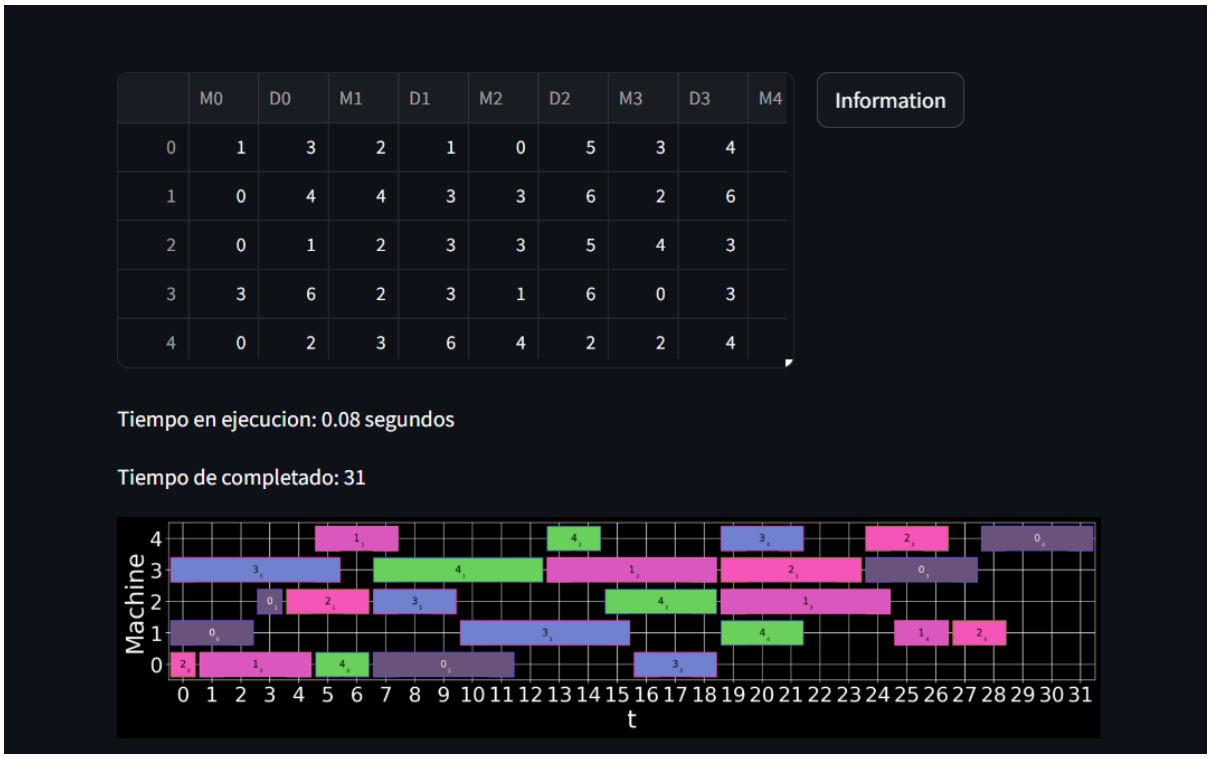
Una solución del problema tiene un aspecto



Podemos tener casos en los cuales cada tarea tiene una duración diferente, como



En esta demo el usuario puede formular un problema JSSP a través de la tabla de instancia, o crear uno aleatorio, y luego utilizar uno de los diversos solver ofrecidos para resolverlo, obteniendo su tiempo de completado del conjunto de trabajos, el tiempo de ejecución del solver, y la gráfica de la solución para casos con tiempo de completado inferior a 60 unidades de tiempo (debido al tamaño de la misma).



Antes de nada, el usuario debe escoger un método de resolución del problema. Los disponibles son:

- Tensor Networks (TN): método quantum-inspired que utiliza tensor networks para simular un sistema cuántico sometido a evolución en tiempo imaginario y técnicas avanzadas de optimización con restricciones, combinando iteración y heurística. Puede resolver cualquier tamaño y forma de problema, funciona rápido y ofrece los mejores resultados para instancias medianas y grandes. No necesita hardware cuántico.
- Tensor Networks Iterativo (TNI): el mismo método de tensor networks, pero iterando sobre un hiperparámetro en vez de utilizar uno predefinido. Tarda 7 veces más, pero en determinados casos ofrece resultados mejores.
- Adaptative: método de resolución de una formulación QUBO específica para el problema. Limitado a problemas de hasta 4x4 con duración de 1 a 3. No usa hardware cuántico.
- TN→Adaptative: método híbrido que primero resuelve con la Tensor Network, obtiene un tiempo de completado y lo usa para ajustar el método Adaptative para buscar una mejor solución. Limitado a problemas de hasta 6x6 con duración de 1 a 3. No usa hardware cuántico.

- TNI→Adaptative: método que hibrida el método iterativo en tensor networks con el Adaptativo, igual que el TN→Adaptative. Limitado a problemas de hasta 6x6 con duración de 1 a 3. No usa hardware cuántico.
- Constrained Quadratic Model (CQM): método de quantum annealing especializado para problemas combinatorios cuadráticos con restricciones. Requiere hardware cuántico, concretamente quantum annealers de Dwave. Limitado a problemas de hasta 10x10 con duración de 1 a 10.
- Binary Quadratic Models (BQM): método de quantum annealing especializado para problemas combinatorios cuadráticos binarios. Requiere hardware cuántico, concretamente quantum annealers de Dwave. Limitado a problemas de hasta 7x7 con duración de 1.
- TN→BQM: híbrido de TN con BQM, como en el caso TN→Adaptative. Requiere hardware cuántico, concretamente quantum annealers de Dwave. Limitado a problemas de hasta 7x7 con duración de 1.
- TNI→BQM: híbrido de TNI con BQM, como en el caso TNI→Adaptative. Requiere hardware cuántico, concretamente quantum annealers de Dwave. Limitado a problemas de hasta 7x7 con duración de 1.

Para formular un problema, el usuario debe primero seleccionar cuantos trabajos y tareas por trabajo desea. Cada método tiene sus propias limitaciones de tamaño. Los métodos TN y TNI están limitados a mano, pero pueden resolver cualquier tamaño y tipo de problema JSSP.

Tras ello, se generará la tabla de instancia, donde cada fila indica un trabajo, y cada pareja de columnas indican la máquina asociada a una tarea y su duración. Todas estas entradas deben ser números enteros. Si un trabajo tiene menos tareas que el número de tareas que hemos indicado, solo debemos añadir duración 0 en las demás tareas.

El usuario también puede optar por generar un caso aleatorio de dicho tamaño o introducir uno a través de un csv. En el caso aleatorio, indicará cuales son las duraciones mínima y máxima posibles para las tareas.

Tras ello, el usuario solo tiene que pulsar en calcular y obtendrá una solución. El tiempo de ejecución depende del método utilizado para la resolución, al igual que la calidad de la solución.