

Trabajo (Semestre 2025-1)
Tópicos en Investigación de Operaciones

Problema:

La empresa GALEM de manufactura está programando las tareas que deben realizar sus máquinas dentro de un taller de pedidos. La empresa cuenta con 3 máquinas disponibles, las cuales pueden realizar cualquiera de los pedidos de los clientes. Al inicio del día cuentan con 30 órdenes de manufactura. Si bien todas las máquinas están capacitadas para realizar los pedidos, cada una opera con un tiempo distinto, dependiendo de la orden que esté trabajando. El objetivo del problema es lograr obtener una programación de la producción que minimice el tiempo requerido para terminar todas las órdenes, ya que esto permite ahorrar en costos de arriendo y gastos de energía.

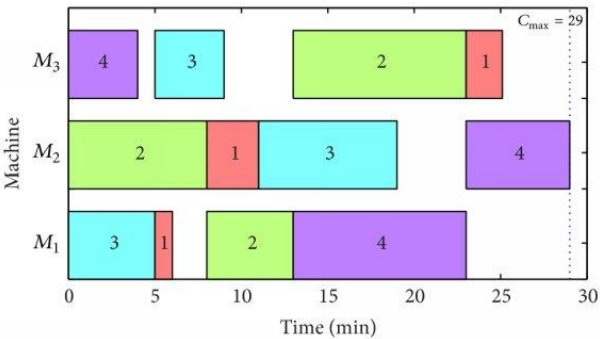
Ustedes tendrán un archivo de Python que contendrá los tiempos de procesos de cada orden en cada máquina. La matriz contenida se puede leer como el siguiente ejemplo:

12	45	78	14	2	9	6	15
61	74	45	150	1	12	11	35
98	72	49	97	55	67	71	84

Los tiempos se leerían:

- El tiempo de la orden 1 en la máquina 1 es de 12 minutos, en la máquina 2 es de 45 minutos y en la máquina 3 es de 78 minutos.
- El tiempo de la orden 3, en la máquina 1 es de 98 minutos, en la máquina 2 es de 72 minutos y en la máquina 3 es de 49 minutos.

En la siguiente figura se puede ver un ejemplo de una solución del problema a resolver. Se debe considerar que en el problema que deben realizar, las órdenes de trabajo no se deben repetir, sino que solo aparecen una vez cada una. El tiempo obtenido de la solución fueron 29 minutos, dado que fue el tiempo de la máquina que más demoró en terminar sus órdenes.



Consejos a considerar:

- 1- Una solución inicial factible es cualquiera que posea todos los trabajos. No importa si utilizan todas las maquinas, alguno o solo una.
- 2- Mientras la solución muestre una vez (no más ni menos) cada orden, la solución será factible. Sin importar el orden.
- 3- El tiempo requerido de la empresa sería igual a la maquina que termina en último lugar.
- 4- Los problemas de este tipo son conocidos como “**job-shop scheduling problem**”.
- 5- Los modelos de la literatura son mucho más complejos que la situación presentada, por lo que no servirá copiar un modelo directamente. La idea del trabajo es que aborden el modelo, lo más parecido a la situación presentada.

A continuación, se presentan una serie de funciones que podría ayudar a la realización del problema. Todas se aplican a listas de Python.

Función / Método	Descripción
sum(lista)	Retorna la suma de todos los elementos numéricos de la lista.
lista.count(valor)	Devuelve el número de veces que aparece un valor en la lista.
max(lista)	Retorna el valor máximo de la lista.
min(lista)	Retorna el valor mínimo de la lista.
lista.append(valor)	Agrega un valor al final de la lista.
lista.insert(posición, valor)	Inserta un valor en la posición indicada de la lista.
lista.pop(posición)	Elimina y devuelve el elemento en la posición dada
random.choice(lista)	Retorna un elemento aleatorio de la lista.
random.randint(inicio, fin)	Retorna un entero aleatorio entre inicio y fin (ambos incluidos).
random.shuffle(lista)	Mezcla aleatoriamente los elementos de la lista (modifica la original).