

# Uso de Q-learning como alternativa a la solución del problema de programación de taller de trabajo flexible

Lic. Arnoldo Del Toro Peña


Facultad de Ingeniería Mecánica y Eléctrica  
Universidad Autónoma de Nuevo León

4 de junio de 2022


# Definiciones

- 1 Agente
- 2 Aprendizaje por refuerzo
- 3 Inteligencia artificial
- 4 Python
- 5 Red neuronal
- 6 Q-learning
- 7 Problema de programación de taller de trabajo flexible.

# Introducción



Metahuerística



Otros modelos  
matemáticos

# Introducción

## Objetivos

¿Es viable utilizar la búsqueda aprendizaje por refuerzo (Q-learning) en el problema de programación de taller de trabajo flexible?



# Introducción

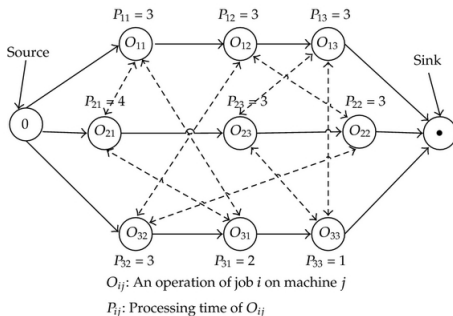
## Objetivos

¿Es viable utilizar la búsqueda aprendizaje por refuerzo (Q-learning) en el problema de programación de taller de trabajo flexible?

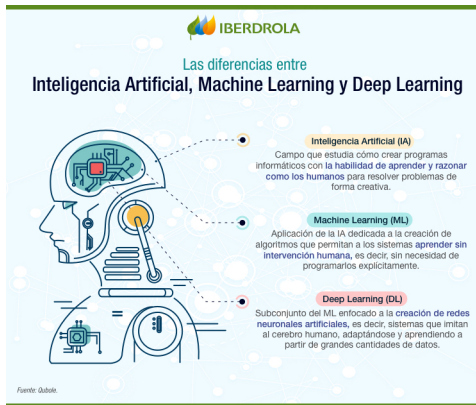
¿Bajo que condiciones es favorable utilizar una búsqueda aprendizaje por refuerzo (Q-learning)?



# Problema de programación de taller de trabajo flexible



# Introducción



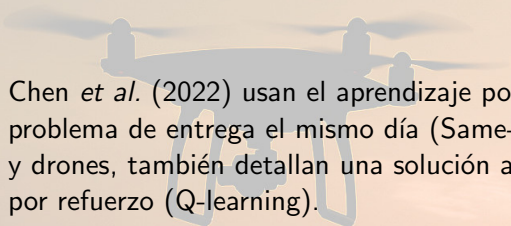
# Obstáculo

Tiempo y recursos.



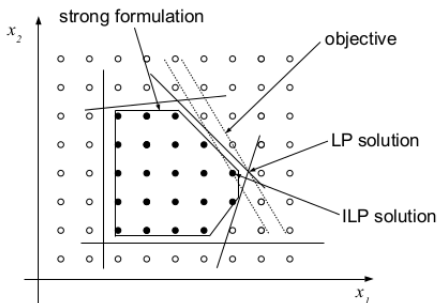


# Antecedentes



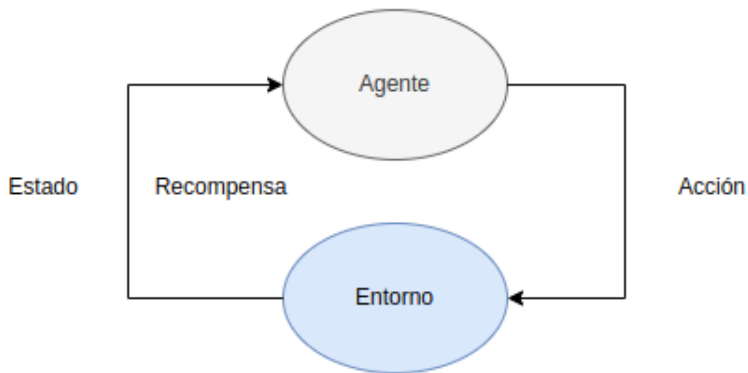
Chen *et al.* (2022) usan el aprendizaje por refuerzo (Q-learning) en un problema de entrega el mismo día (Same-day delivery) utilizando vehículos y drones, también detallan una solución aproximada usando el aprendizaje por refuerzo (Q-learning).

Huang *et al.* (2022) presentan el método clasificación de corte (Cut Ranking) para seleccionar los cortes en un problema de ramificación y corte para programación entera mixta (mixed-integer programming).



Zhao *et al.* (2019) proponen la implementación de un algoritmo de doble capa de aprendizaje por refuerzo (Q-learning) así como acciones para la solución del problema dinámico de programación de talleres flexibles con fallas en máquinas.

## Metodología



# Metodología

(Continuación)

Los pasos específicos están detallados a continuación:

- Inicializar  $Q(s, a)$  arbitrariamente
- Establecer el parámetro  $\gamma$  y  $\alpha$
- Repetir (para cada episodio):
  - ① Inicializar  $s$
  - ② Repetir (para cada paso del episodio):
    - 2.1 Elegir  $a$  desde  $s$  usando una de las políticas de  $Q$ .
    - 2.2 Tomar la acción  $a$ , observar  $r, s'$
    - 2.3  $Q(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma \max_{a'}(Q(s', a') - Q(s, a))]$
    - 2.4  $s \leftarrow s'$
    - 2.5 Hasta que  $s'$  sea terminal.

# Metodología

(Continuación)

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha[r_{t+1} + \gamma \max_a (Q(s_{t+1}, a) - Q(s_t, a_t))] \quad (1)$$

Donde  $r_t$  representa la recompensa recibida cuando el Agente es transferido del estado  $s_t$  al estado  $s_{t+1}$ . Y  $\alpha$  representa la tasa de aprendizaje ( $\alpha \in (0, 1]$ ).

# Metodología

(Continuación)

El objetivo del algoritmo aprendizaje por refuerzo (Q-learning) se actualiza siguiendo la fórmula:

$$r_{t+1} + \gamma \max Q(s_{t+1}, a) \quad (2)$$

Y un episodio del algoritmo termina cuando el estado  $s_{t+1}$  es terminal.

# Metodología

(Continuación)

En la primer capa el conjunto de acciones son las siguientes (Zhao *et al.*, 2019):

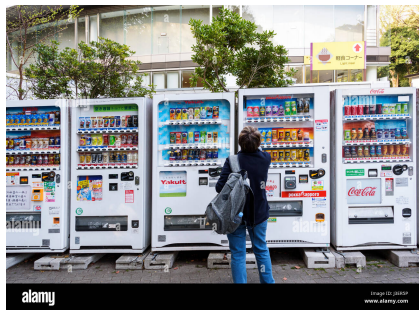
- 1 SPT: representa el mínimo de tiempo de procesamiento de las operaciones que serán seleccionados
- 2 EDD: representa el mínimo de tiempo de entrega en las operaciones que serán seleccionadas
- 3 FIFO: representa el primer trabajo en llegar que será seleccionado.



# Metodología

## Continuación

La segunda capa (Zhao *et al.*, 2019).



# Metodología

(Continuación)

Estados (state).



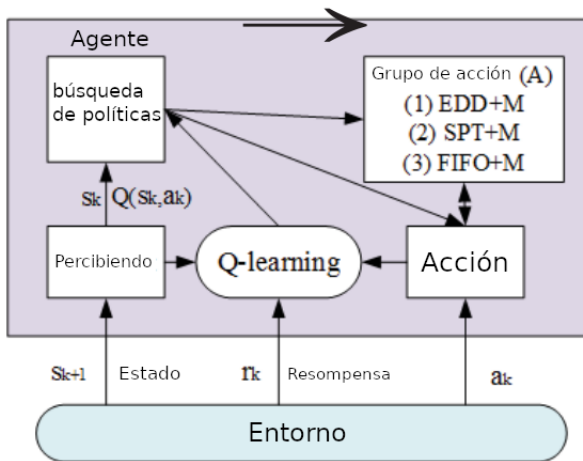
$$SD = \frac{100 \times T}{RT} \quad (3)$$

# Recompensa

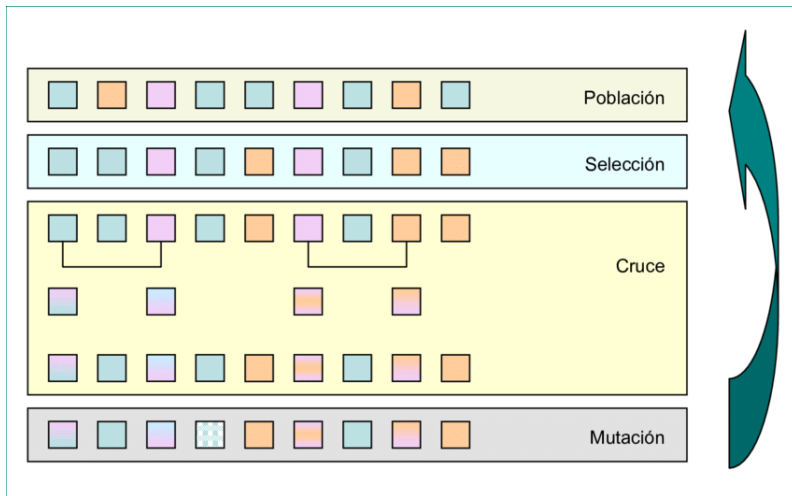
“Entre más tiempo de espera menos es la recompensa”



# Estructura



# Algoritmo Genético (Batista *et al.*, 2009)



# Planificación con base en aprendizaje por refuerzo (Q-learning)

Por ejemplo:

- ① Ninguna
- ② SPT+M
- ③ EDD+M
- ④ FIFO+M

# Conclusiones

- 
- 1 problema de programación de taller de trabajo flexible
  - 2 tiempo
  - 3 resultados.

# References I

- Batista, M., Moreno-Pérez, J., y Moreno-Vega, J. (2009). Algoritmos genéticos. una visión práctica. *Números, ISSN 0212-3096, N<sup>o</sup>. 71, 2009 (Ejemplar dedicado a: Darwin)*.
- Chen, X., Ulmer, M. W., y Thomas, B. W. (2022). Deep q-learning for same-day delivery with vehicles and drones. *European Journal of Operational Research*, 298(3):939–952.
- Huang, Z., Wang, K., Liu, F., Zhen, H.-L., Zhang, W., Yuan, M., Hao, J., Yu, Y., y Wang, J. (2022). Learning to select cuts for efficient mixed-integer programming. *Pattern Recognition*, 123:108353.
- Zhao, M., Li, X., Gao, L., Wang, L., y Xiao, M. (2019). An improved q-learning based rescheduling method for flexible job-shops with machine failures. En *2019 IEEE 15th international conference on automation science and engineering (CASE)*, pp. 331–337. IEEE.