



rubik-cube

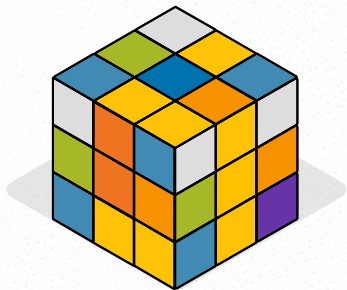
*Resolución de cubos Rubik mediante
Inteligencia Artificial*

Cubo Rubik

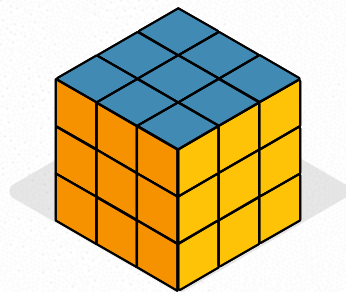
Definición del Problema

Cubo Rubik

Estado Inicial



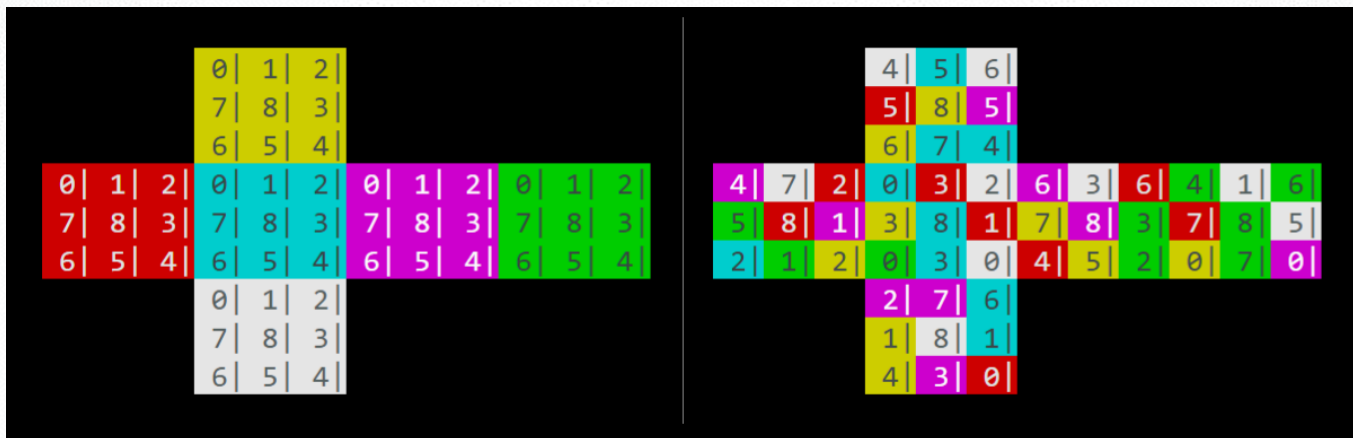
Solución



- ▶ 26 piezas tridimensionales
- ▶ 12 acciones/rotaciones posibles
- ▶ Medida de fitness: cantidad de piezas colocadas correctamente

- ▶ 43 trillones de estados posibles
- ▶ 1 estado objetivo

Implementación



- ▶ 6 atributos de cara
- ▶ 9 atributos por cara para cada sub-pieza
- ▶ 12 métodos de acción/rotación

- ▶ Historial de acciones
- ▶ Función fitness

Algoritmos utilizados

para resolver el problema

Algoritmo Genético

- ▶ Operadores de Selección Natural: **selección, entrecruzamiento, mutación.**
- ▶ Cada **ejecución** busca **resolver** un cubo en un **estado inicial específico.**
- ▶ **Población inicial:** mezclas aleatorias.
- ▶ **Selección de padres:** probabilidad según fitness.
- ▶ **Entrecruzamiento:** single-point crossover.
- ▶ **Mutación:** con alta probabilidad en 3 puntos.
- ▶ **Selección:** individuos más aptos entre padres e hijos.

Deep Reinforcement Learning

- ▶ Reinforcement Learning + Deep Learning
- ▶ **Modelo:** Deep Q Network
- ▶ **Agente:** Train DQN Algorithm
- ▶ **Entorno:** Cubo Rubik
- ▶ **Objetivo:** maximizar la **suma de recompensas** tomando **acciones** a través de **episodios**

Algoritmo Completamente Aleatorio

- ▶ Implementado para **medir** la eficiencia de los otros métodos.
- ▶ **Funcionamiento**: aplicar **20 acciones aleatorias distintas** en **25000** cubos iguales y registrar el **mejor resultado**.

Métodos de prueba

de los algoritmos

Métodos de prueba

- ▶ **Deep Reinforcement Learning:** no fue considerado. Realiza siempre la misma acción.
- ▶ **Algoritmo Genético:**
 - Paso 1:
 - 7 distintas combinaciones de parámetros
 - 30 distintos estados iniciales
 - Se elige la combinación de parámetros más efectiva

Métodos de prueba

► Algoritmo Genético

- Paso 2:

- 135 distintos estados iniciales
- Combinación de parámetros:
 - Número máximo de generaciones: 100
 - Individuos por generación: 250
 - Probabilidad de mutación: 0.5
 - Split percent: 0.2
 - Acciones iniciales: 30
 - Hijos por pareja de padres: 2

► Algoritmo Aleatorio:

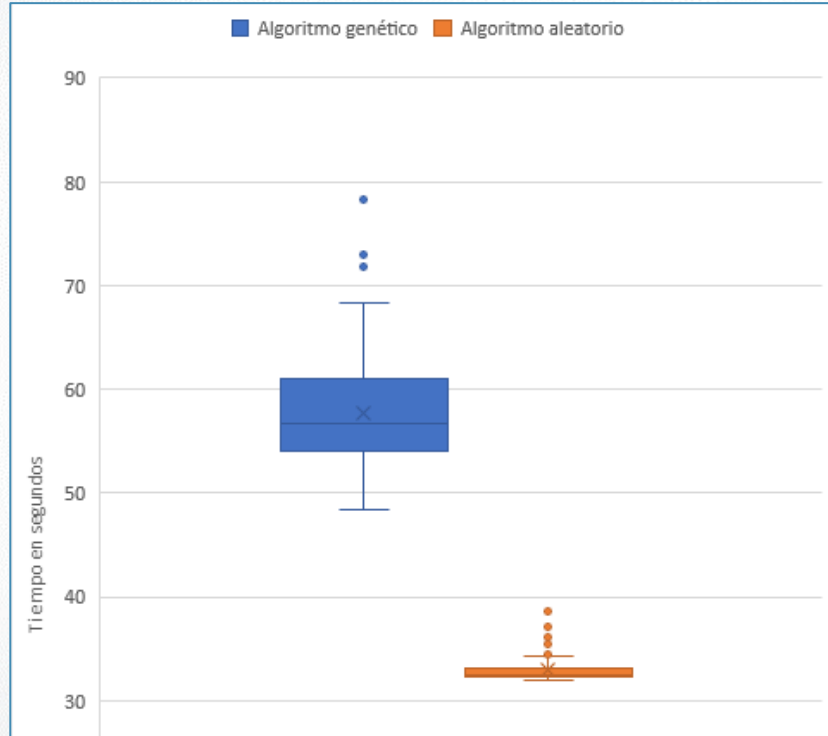
- 135 distintos estados iniciales

► **Método de medición:** ejecutar el algoritmo genético y aleatorio sobre los mismos 135 estados iniciales.

Resultados

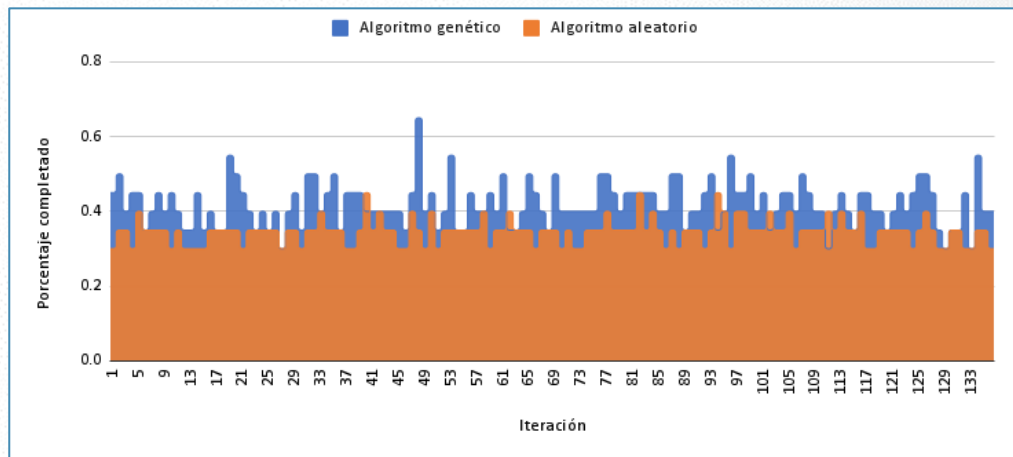
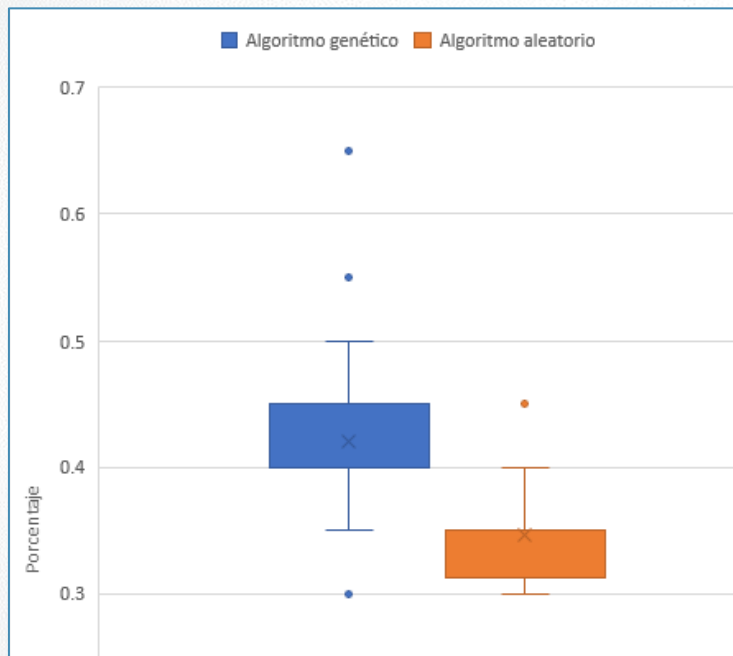


Tiempos de ejecución



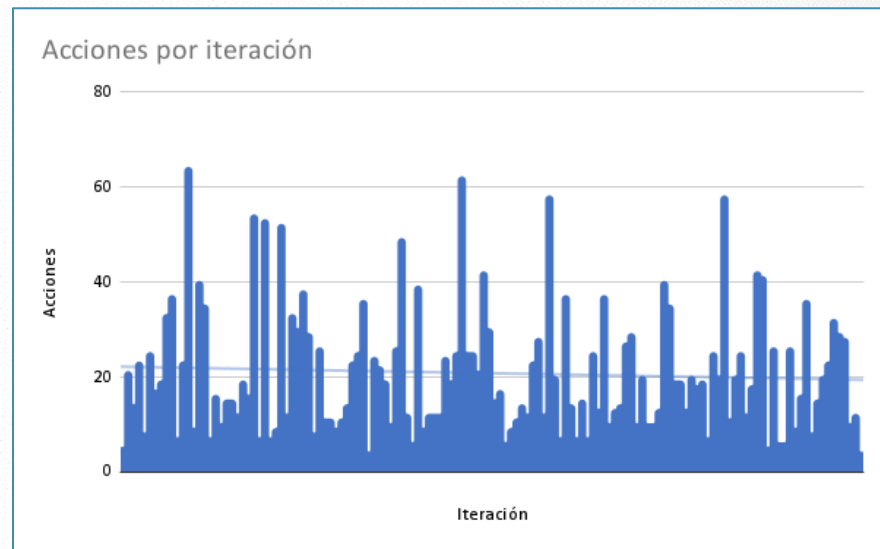
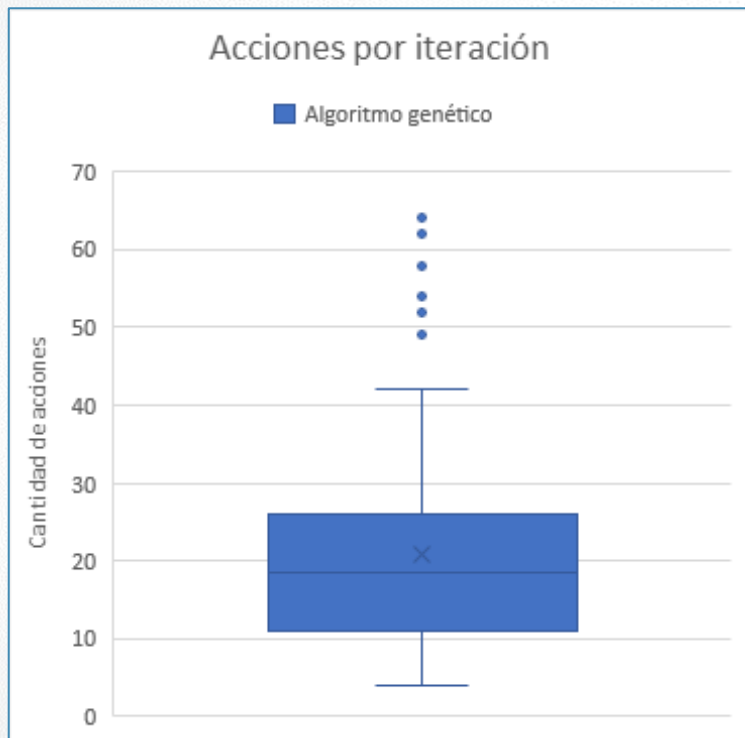
Algoritmo Aleatorio **57%** más rápido que Algoritmo Genético

Porcentaje completado



Algoritmo Genético **8%** más eficaz que Algoritmo Aleatorio

Acciones por iteración (Algoritmo Genético)



Promedio: 21 acciones

Conclusiones

e ideas futuras

Conclusiones

- ▶ Ningún algoritmo llega al estado objetivo.
- ▶ El Algoritmo Genético obtiene 8% mejores resultados que el Algoritmo Aleatorio.
- ▶ El Algoritmo Aleatorio tiene 57% menor costo temporal que el Algoritmo Genético.
- ▶ El Algoritmo Genético genera soluciones con 21 acciones en promedio.
- ▶ El Algoritmo de DRL no logra ningún resultado superior al 10% de progreso.

Ideas Futuras

- ▶ Realizar mejoras en el **Algoritmo Aleatorio** para mejorar su **efectividad**.
- ▶ Aplicar restricciones basadas en **teoría de grupos** a las **secuencias** de acciones aleatorias.
- ▶ **Reimplementar** el algoritmo de **Deep Reinforcement Learning**.

A decorative background graphic consisting of a network of interconnected nodes and lines. The nodes are represented by small colored circles in blue, orange, teal, and purple. Some nodes are highlighted with concentric dashed circles. The lines are thin and grey, forming a complex web-like structure across the slide.

¿Preguntas?

GRACIAS