



Instituto Tecnológico de Costa Rica

Escuela de Ingeniería en Computación

IC-6200 – Inteligencia Artificial

Tarea Corta 2 – Algoritmos Genéticos

Estudiantes:

Pablo Mauricio Mesén Alvarado – 2023071259

Samir Fernando Cabrera Tabash – 2022161229

Luis Gerardo Urbina Salazar – 2023156802

Profesor:

Kenneth Roberto Obando Rodríguez

24 de septiembre de 2025

II Semestre 2025

Índice

1. Resumen ejecutivo	2
2. Introducción	2
3. Descripción de la modalidad escogida (juego)	3
4. Diseño del cromosoma y función de fitness	4
4.1. Representación del cromosoma	4
4.2. Política de control	4
4.3. Función de fitness	4
5. Parámetros y configuración experimental	5
5.1. Parámetros del algoritmo genético	5
5.2. Operadores evolutivos	5
5.3. Configuración experimental	6
6. Resultados y visualizaciones	6
7. Discusión crítica y análisis	9
8. Conclusiones y trabajo futuro	10

1. Resumen ejecutivo

Este trabajo implementa un algoritmo genético (AG) para resolver el problema de control de CartPole en el entorno Gymnasium. Se evaluaron 27 configuraciones experimentales combinando tamaños de población (10, 20, 30 individuos), números de generaciones (10, 30, 50) y tres métodos de crossover.

La metodología emplea cromosomas de 4 genes (pesos) correspondientes a las variables de estado del CartPole. La función de fitness se basa en el número de pasos que el agente mantiene el equilibrio (máximo 500). Se implementó paralelización para optimizar tiempos de evaluación, selección elitista del 20

Los resultados muestran que múltiples configuraciones alcanzaron el fitness máximo de 500 pasos. Las configuraciones con mayor número de generaciones (50) y poblaciones más grandes (30) presentaron mayor consistencia en la convergencia. Los algoritmos genéticos resultaron efectivos para este problema de control, proporcionando una alternativa viable a métodos tradicionales de aprendizaje por refuerzo.

2. Introducción

Los algoritmos genéticos son técnicas de optimización bioinspiradas basadas en los principios de la evolución natural. Utilizan mecanismos como selección, crossover y mutación para evolucionar poblaciones de soluciones candidatas hacia óptimos globales.

El problema de CartPole consiste en mantener equilibrado un poste articulado sobre un carro que se mueve horizontalmente. El agente debe aplicar fuerzas hacia la izquierda o derecha para evitar que el poste caiga. Este problema es un benchmark clásico en aprendizaje por refuerzo debido a su simplicidad conceptual pero desafío de control continuo.

Los algoritmos genéticos ofrecen ventajas para problemas de control: no requieren información sobre gradientes, pueden manejar espacios de búsqueda discontinuos, y son naturalmente paralelizables. En el contexto de CartPole, permiten evolucionar políticas de control representadas como combinaciones lineales de variables de estado.

La motivación de este trabajo es evaluar la efectividad de los algoritmos genéticos en un problema de control clásico, comparando diferentes configuraciones de parámetros y métodos de crossover. Los fundamentos teóricos incluyen la codificación de políticas como cromosomas, la definición de fitness basada en rendimiento temporal, y la aplicación de operadores evolutivos para la mejora iterativa de las soluciones.

3. Descripción de la modalidad escogida (juego)

CartPole-v1 es un entorno de control clásico disponible en la biblioteca Gymnasium (anteriormente OpenAI Gym). El problema consiste en balancear un poste articulado sobre un carro que se desplaza horizontalmente sobre una pista de longitud limitada.

Cart Pole



Figura 1: Imagen de referencia a CartPole.

Espacio de estados: El estado del sistema se representa mediante un vector de 4 dimensiones:

- Posición del carro: $x \in [-4, 8, 4, 8]$
- Velocidad del carro: $\dot{x} \in (-\infty, \infty)$
- Ángulo del poste: $\theta \in [-24^\circ, 24^\circ]$ (aproximadamente)
- Velocidad angular del poste: $\dot{\theta} \in (-\infty, \infty)$

Espacio de acciones: El agente puede aplicar una fuerza discreta:

- Acción 0: Empujar el carro hacia la izquierda
- Acción 1: Empujar el carro hacia la derecha

Función de recompensa: El agente recibe una recompensa de +1 por cada paso temporal en que mantiene el poste equilibrado. Un episodio termina cuando:

- El ángulo del poste excede los $\pm 12^\circ$ de la vertical
- El carro se sale de los límites de la pista (± 2.4 unidades)
- El episodio alcanza 500 pasos (considerado como éxito)

La dificultad del problema radica en la naturaleza inestable del sistema: el poste tiende naturalmente a caer, requiriendo acciones de control continuas para mantener el equilibrio. El desafío es aprender una política que balance efectivamente las fuerzas aplicadas al carro para estabilizar el poste mediante retroalimentación basada únicamente en el estado observable del sistema.

4. Diseño del cromosoma y función de fitness

4.1. Representación del cromosoma

Cada individuo en la población se representa como un vector de pesos $\mathbf{w} = [w_1, w_2, w_3, w_4]$, donde cada componente corresponde a una variable de estado del CartPole:

- w_1 : peso para la posición del carro
- w_2 : peso para la velocidad del carro
- w_3 : peso para el ángulo del poste
- w_4 : peso para la velocidad angular del poste

Los pesos se inicializan aleatoriamente en el rango $[-1, 1]$ y se mantienen acotados en $[-2, 2]$ durante la evolución para evitar valores extremos.

4.2. Política de control

La política implementada es una función lineal simple:

$$\text{acción} = \begin{cases} 0 & \text{si } \mathbf{w} \cdot \mathbf{s} < 0 \\ 1 & \text{si } \mathbf{w} \cdot \mathbf{s} \geq 0 \end{cases} \quad (1)$$

donde \mathbf{s} es el vector de estado observado y $\mathbf{w} \cdot \mathbf{s}$ representa el producto punto entre los pesos y el estado.

4.3. Función de fitness

La función de fitness evalúa el rendimiento promedio de un individuo ejecutando múltiples episodios:

$$\text{fitness}(\mathbf{w}) = \frac{1}{N} \sum_{i=1}^N R_i(\mathbf{w}) \quad (2)$$

donde $N = 3$ es el número de episodios de evaluación y $R_i(\mathbf{w})$ es la recompensa total obtenida en el episodio i (número de pasos hasta la terminación, máximo 500).

5. Parámetros y configuración experimental

5.1. Parámetros del algoritmo genético

- **Tamaño de población:** 10, 20, 30 individuos
- **Número de generaciones:** 10, 30, 50
- **Tasa de mutación:** 0.1 (10%)
- **Episodios por evaluación:** 3
- **Método de selección:** Elitismo (20% de supervivientes)

5.2. Operadores evolutivos

Mutación gaussiana: Se aplica ruido gaussiano con media 0 y desviación estándar 0.2:

$$\mathbf{w}' = \mathbf{w} + \mathcal{N}(0, 0.2^2) \quad (3)$$

Métodos de crossover implementados:

1. **Crossover uniforme:** Para cada gen, se selecciona aleatoriamente el valor del padre 1 o padre 2 con probabilidad 0.5:

$$\text{hijo}[i] = \begin{cases} \text{padre}_1[i] & \text{si } \text{rand}() < 0.5 \\ \text{padre}_2[i] & \text{caso contrario} \end{cases} \quad (4)$$

2. **Crossover de un punto:** Se selecciona un punto de corte aleatorio $k \in [1, 3]$ y se intercambian los segmentos:

$$\text{hijo} = [\text{padre}_1[0 : k], \text{padre}_2[k : 4]] \quad (5)$$

3. **Crossover de dos puntos:** Se seleccionan dos puntos de corte $k_1 < k_2$ y se intercambia el segmento intermedio:

$$\text{hijo} = [\text{padre}_1[0 : k_1], \text{padre}_2[k_1 : k_2], \text{padre}_1[k_2 : 4]] \quad (6)$$

5.3. Configuración experimental

Se ejecutaron 27 combinaciones experimentales (3 tamaños de población \times 3 números de generaciones \times 3 métodos de crossover). La implementación utiliza paralelización con $n - 1$ procesos (donde n es el número de cores disponibles) para acelerar la evaluación de fitness. Cada configuración genera gráficos comparativos y guarda el mejor individuo en formato JSON.

6. Resultados y visualizaciones

Los resultados experimentales se presentan mediante gráficos que muestran la evolución del fitness promedio y máximo a lo largo de las generaciones para diferentes configuraciones de población y generaciones. Cada gráfico compara los tres métodos de crossover implementados (uniforme, un punto y dos puntos) bajo las mismas condiciones experimentales.

Las visualizaciones permiten analizar la convergencia del algoritmo, identificar configuraciones óptimas y evaluar la efectividad relativa de los diferentes operadores de crossover. Los gráficos muestran tanto la tendencia central del fitness poblacional como los valores máximos alcanzados, proporcionando una visión completa del comportamiento evolutivo.

Se observa que configuraciones con mayor número de generaciones tienden a alcanzar mayor estabilidad en el fitness, mientras que poblaciones más grandes ofrecen mayor diversidad genética inicial. Los patrones de convergencia varían según el método de crossover utilizado, siendo algunos más efectivos para ciertas combinaciones de parámetros.

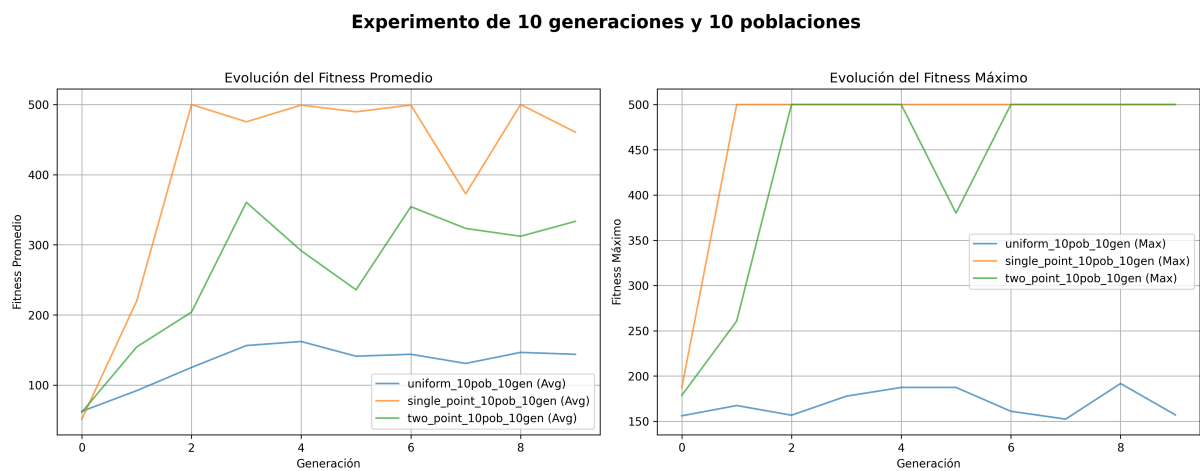


Figura 2: Resultados de entrenamiento del CartPole con configuración de 10 generaciones y 10 poblaciones.

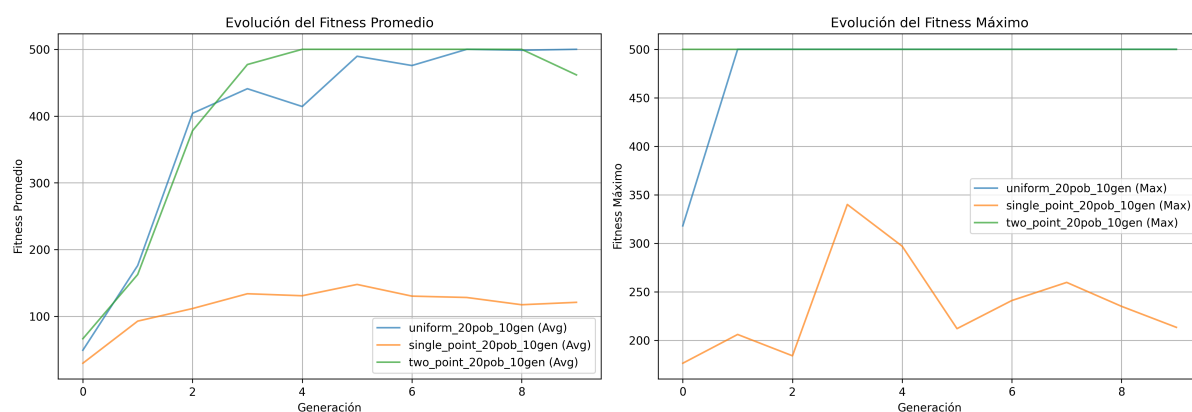
Experimento de 10 generaciones y 20 poblaciones

Figura 3: Resultados de entrenamiento del CartPole con configuración de 10 generaciones y 20 poblaciones.

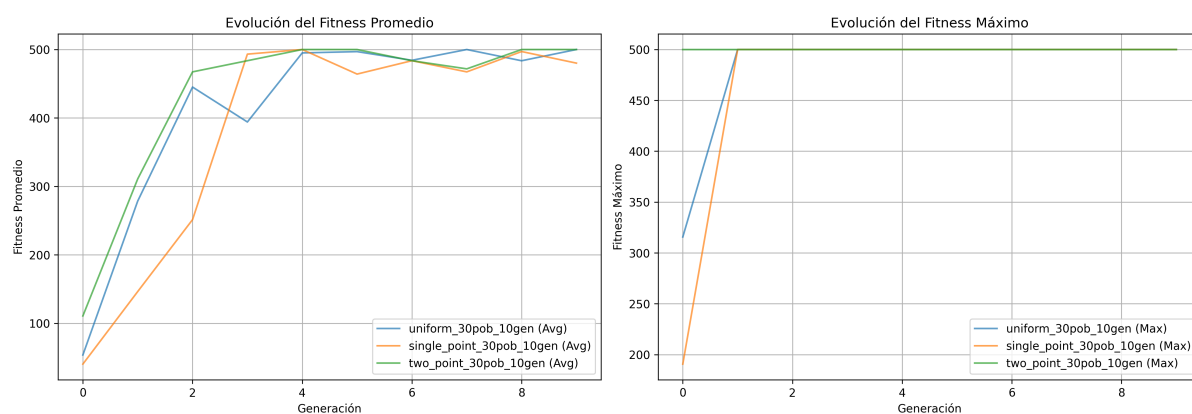
Experimento de 10 generaciones y 30 poblaciones

Figura 4: Resultados de entrenamiento del CartPole con configuración de 10 generaciones y 30 poblaciones.

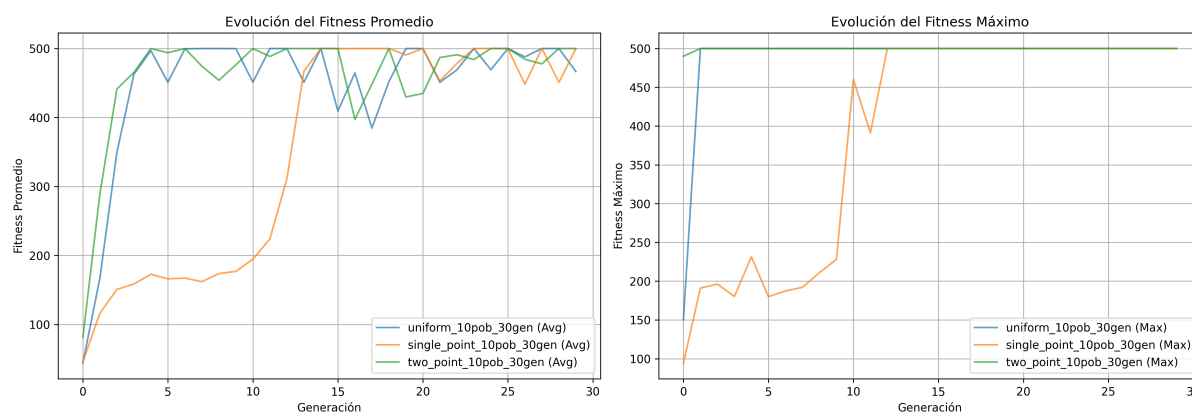
Experimento de 30 generaciones y 10 poblaciones

Figura 5: Resultados de entrenamiento del CartPole con configuración de 30 generaciones y 10 poblaciones.

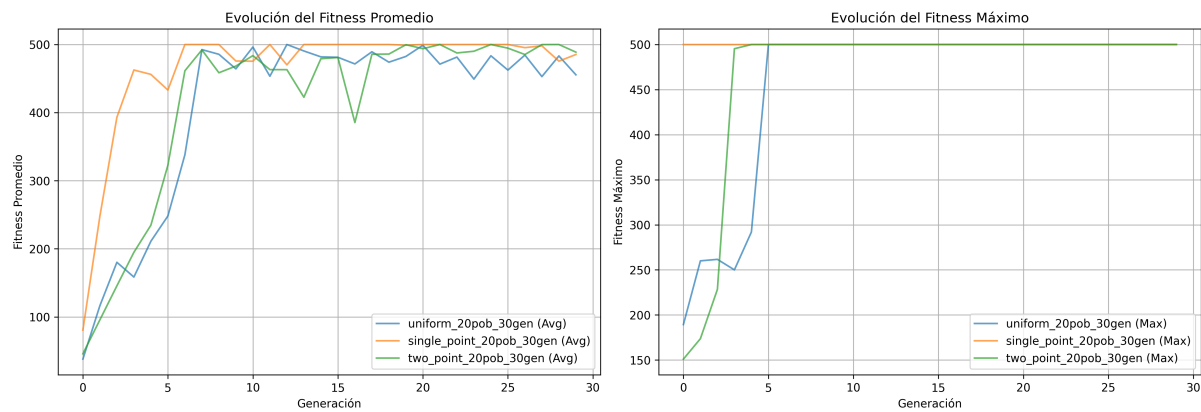
Experimento de 30 generaciones y 20 poblaciones

Figura 6: Resultados de entrenamiento del CartPole con configuración de 30 generaciones y 20 poblaciones.

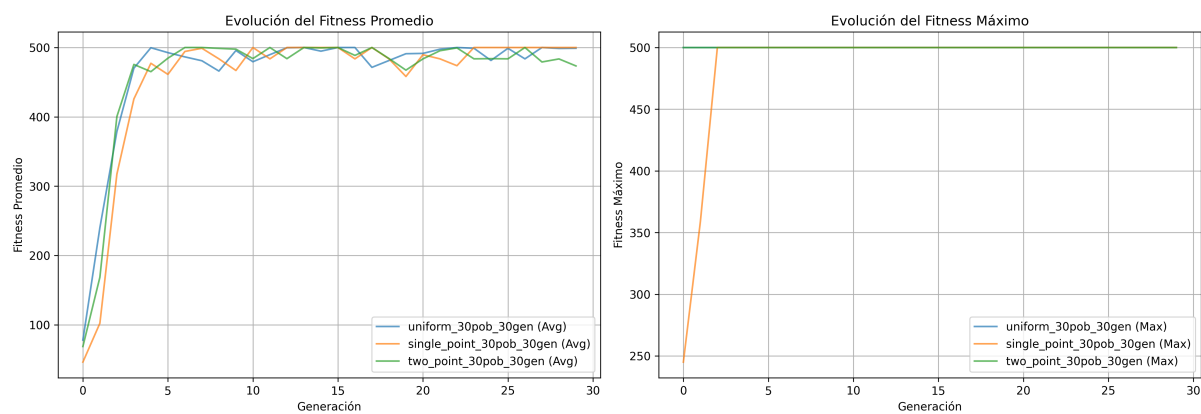
Experimento de 30 generaciones y 30 poblaciones

Figura 7: Resultados de entrenamiento del CartPole con configuración de 30 generaciones y 30 poblaciones.

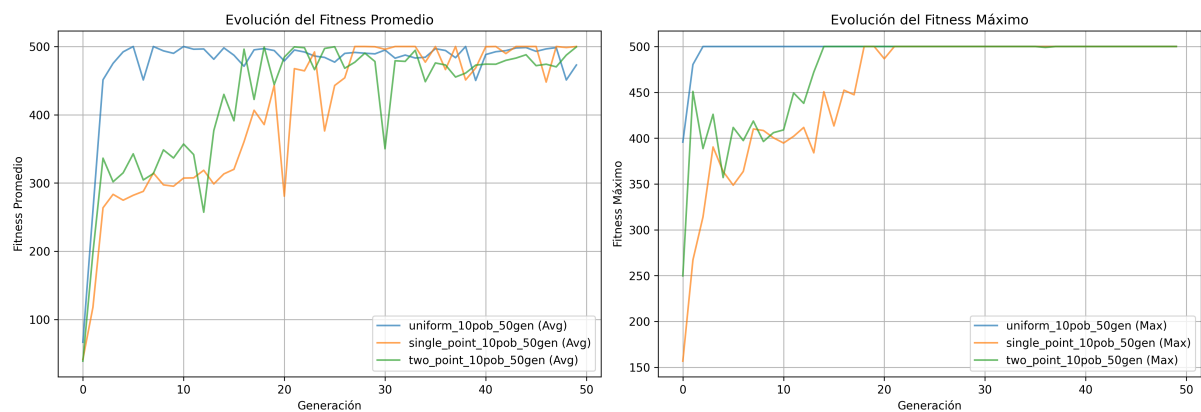
Experimento de 50 generaciones y 10 poblaciones

Figura 8: Resultados de entrenamiento del CartPole con configuración de 50 generaciones y 10 poblaciones.

Experimento de 50 generaciones y 20 poblaciones

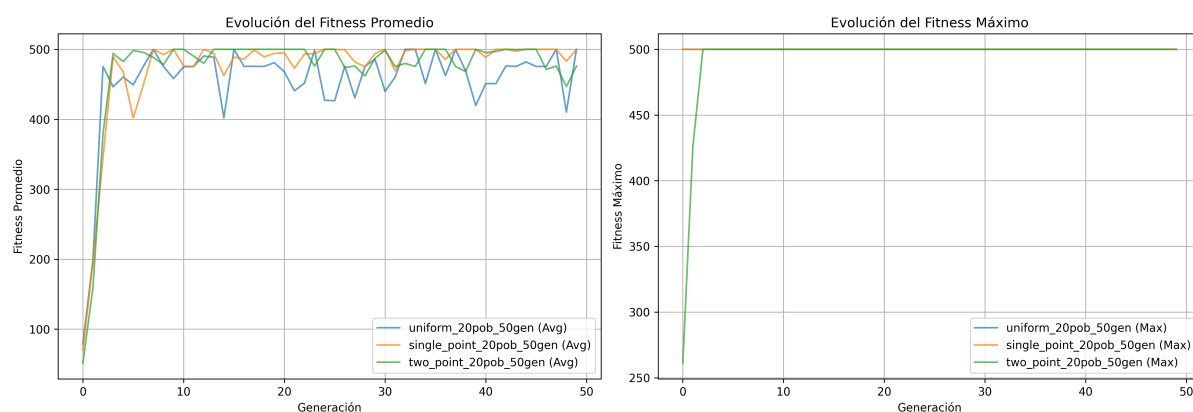


Figura 9: Resultados de entrenamiento del CartPole con configuración de 50 generaciones y 20 poblaciones.

Experimento de 50 generaciones y 30 poblaciones

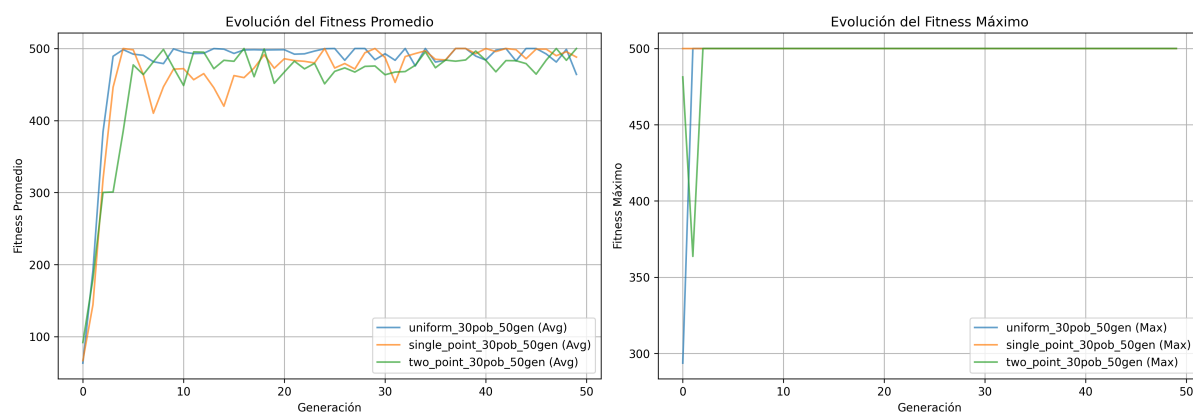


Figura 10: Resultados de entrenamiento del CartPole con configuración de 50 generaciones y 30 poblaciones.

7. Discusión crítica y análisis

Los resultados experimentales revelan que múltiples configuraciones lograron alcanzar el fitness máximo de 500 pasos, indicando convergencia exitosa hacia políticas de control efectivas. Sin embargo, alcanzar el máximo puntaje no es el único criterio de éxito, ya que existen diferencias cualitativas en el comportamiento del agente.

El análisis crítico debe considerar varios aspectos:

- **Estabilidad del control:** Aunque varios modelos alcanzaron 500 pasos, difieren en la suavidad y estabilidad de sus movimientos. Algunos agentes mantienen el

poste equilibrado con movimientos mínimos y eficientes, mientras otros requieren correcciones constantes y bruscas.

- **Robustez:** La evaluación mediante múltiples episodios ayuda a identificar políticas consistentes versus aquellas que funcionan esporádicamente. Modelos con alta varianza entre episodios sugieren soluciones menos robustas.
- **Eficiencia computacional:** Las configuraciones con poblaciones más grandes (30 individuos) y más generaciones (50) requieren significativamente más tiempo de cómputo, pero no necesariamente producen mejores resultados que configuraciones más modestas.
- **Métodos de crossover:** Los tres métodos implementados muestran diferencias en velocidad de convergencia y exploración del espacio de soluciones. El crossover uniforme tiende a mantener mayor diversidad, mientras que los métodos de punto preservan mejor los bloques de construcción genéticos.

8. Conclusiones y trabajo futuro

Los algoritmos genéticos demostraron ser efectivos para resolver el problema de control de CartPole cuando se configuran apropiadamente. La simplicidad de la representación cromosómica (4 pesos) permite una evolución eficiente hacia políticas de control competentes.

Conclusiones principales

- Configuraciones modestas (20-30 individuos, 30-50 generaciones) son suficientes para alcanzar soluciones óptimas.
- La paralelización reduce significativamente los tiempos de entrenamiento, haciendo factible la experimentación exhaustiva.
- Los diferentes métodos de crossover ofrecen trade-offs entre exploración y explotación del espacio de búsqueda.
- La evaluación multi-episodio mejora la selección de políticas robustas.

Trabajo futuro

- Implementar representaciones cromosómicas más complejas (redes neuronales pequeñas).
- Evaluar algoritmos genéticos en entornos de control más desafiantes.
- Desarrollar métricas de calidad que consideren la eficiencia energética del control.
- Comparar directamente con métodos de aprendizaje por refuerzo clásicos (Q-learning, Policy Gradient).
- Investigar técnicas de co-evolución para balancear múltiples objetivos de control.